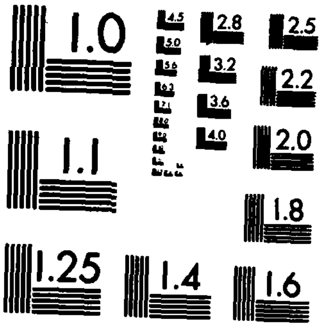
AD-A147 105 NICHOLET MED-80 (TRADEMARK) PROGRAMS IN THE NAVAL BIODYNAMICS LABORATORY EVOKED POTENTIAL SERIES(U) NAVAL BIODYNAMICS LAB NEW ORLEANS LA R E TABLER JUL 84 1/1

UNCLASSIFIED NBDL-84R007 F/G 6/19 NL

END

DTIC


1.0
4.5
5.0
5.6
6.3
7.1
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

NBDL - 84R007

NICOLET MED-80® PROGRAMS IN THE NAVAL BIODYNAMICS LABORATORY EVOKED POTENTIAL SERIES

ROBERT E. TABLER, JR.

JULY 1984

NAVAL BIODYNAMICS LABORATORY
New Orleans, Louisiana

Approved for public release. Distribution unlimited.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NBDL-84R007	2. GOVT ACCESSION NO. AD-A147 105	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) NICHOLET MED-80® PROGRAMS IN THE NAVAL BIODYNAMICS LABORATORY EVOKED POTENTIAL SERIES		5. TYPE OF REPORT & PERIOD COVERED Research Report
		6. PERFORMING ORG. REPORT NUMBER NBDL-84R007
7. AUTHOR(s) Robert E. Tabler, Jr.		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Biodynamics Laboratory P. O. Box 29407 New Orleans, LA 70189		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS M0097PN001-5004 63706N
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Medical Research & Development Command Bethesda, MD 20814		12. REPORT DATE July 1984
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Nicolet Med-80®, Computer Program, Evoked Potentials

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report gives complete documentation and listings for five computer programs developed at the Naval Biodynamics Laboratory to record, analyze, and plot evoked potential data using the Nicolet Med-80® computer. The programs are: EPPROG, PERDAT, EPDATA, PLOTEP, and LSTPER. Appendices include: wiring for the Med-80®, flowcharts for all programs, and printouts of all programs.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

NBDL - 84R007

NICOLET MED-80® PROGRAMS
IN THE NAVAL BIODYNAMICS LABORATORY
EVOKED POTENTIAL SERIES

ROBERT E. TABLER, JR.

July 1984

Naval Medical Research and Development Command
Research Work Unit No. M0097PN001-5004

Approved by	Released by
J. C. GUIGNARD Chairman, Editorial Review Board	Captain L. E. WILLIAMS, MC, USN Commanding Officer

Naval Biodynamics Laboratory
P. O. Box 29407
New Orleans, LA 70189

Opinions or conclusions contained in this report are those of the author and do not necessarily reflect the views or the endorsement of the Department of the Navy. Approved for public release; distribution unlimited. Reproduction in whole or in part is permitted for any purpose of the United States Government.

SUMMARY

PROBLEM:

Average evoked potentials are used to monitor possible dysfunctions in the central nervous system of rhesus monkey (M. mulatta) or human subjects during impact acceleration studies at the Naval Biodynamics Laboratory. Automated procedures are required to collect evoked potential via the Nicolet Med-80® with minimum interaction by the experimenter.

FINDINGS:

This report gives complete documentation and listings for five computer programs developed at the Naval Biodynamics Laboratory to record, analyze, and plot evoked potential data. The programs are EPPROG, PERDAT, EPDATA, PLOTEP, and LSTPER.

The EPPROG program (Evoked Potential) is a central program used as a pointer to all other programs in the evoked potential test series. A short introduction and some general instructions for each program are provided, if requested.

The PERDAT program (Personal Data) is used to collect personal and montage information for the evoked potential series on both human and rhesus subjects.

The EPDATA program (Evoked Potential Data) is used to automatically collect and store parameter values and evoked potential data.

The PLOTEP program (Plot Evoked Potentials) is used to plot both evoked potential parameter and waveform data, with the corresponding personal data on the Nicolet Zeta Plotter®.

The LSTPER program (List Data) is used to provide a detailed listing of the data contained in the personal data file and the parameter data file.

It should be noted that these programs can also be used in a clinical setting, though some modifications of the personal data portions should be made.

RECOMMENDATAIONS:

Any new programs or changes made to the current programs should be incorporated in this manual.

ACKNOWLEDGEMENTS:

Special thanks to Dr. Marc S. Weiss for his technical expertise.

Trade names of materials or products of commercial or non-government organizations are cited where essential for precision in describing research procedures or evaluation of results. Their use does not constitute official endorsement or approval of the use of such commercial hardware or software.

Current address: Robert E. Tabler, Jr., Department of Psychology, University of
West Florida, Pensacola, Florida 32504.

TABLE OF CONTENTS

PROGRAMS IN THE NAVAL BIODYNAMICS LABORATORY EVOKED POTENTIAL SERIES

APPENDIXES

EPPROG

PURPOSE: The EPPROG program (Evoked Potential) is a central program used as a pointer to all other programs in the evoked potential test series. A short introduction and some general instructions for each program are provided, if requested.

10 The variables to be used in the EPPROG program are dimensioned.
10 - 40 The paper is advanced.

50 The user is asked whether or not a listing or a description of the
50 - 220 programs being offered is needed. A 'Y' response to the question will cause some general instructions for the execution of all programs to be printed out. A listing of the programs in the evoked potential series and the file names used in calling up the programs are also provided. An 'N' response to the question will cause the program to advance to line 230.

230 The user inputs the name of the program to be executed, and vali-
230 - 310 dates the response. Via a conditional branch, the EPPROG program is advanced to the location which contains the information on the requested program. The program requested is then executed. The user can enter 'END' here to exit from the program.

320 These lines contain a brief introduction to each of the programs.
320 - 790 Some of the programs also contain the pre-run information needed. This is followed by a RUN statement for that program.

800 End of program.

VARIABLES USED IN EPPROG

PROG$ - selected program to be run
X$ - response to a yes or no question

PERDAT

PURPOSE: The PERDAT program (Personal Data) is used to collect personal and montage information for the evoked potential series on both human and rhesus subjects.

10 All variables to be used in this program are dimensioned.
10 - 50

60 Values are assigned to some variables.
60 - 70

80 The following steps are repeated for each variable.
80 - 710

1. P is set equal to the size of the variable under consideration.
2. The variable under consideration is filled with a series of #'s (see SUBROUTINE 1930).
3. The user is now asked to input the value of the variable. If there is no input for this variable then the return key is pressed causing the #'s to be left in place.
4. XX$ is now made equal to the variable under consideration and the user is asked to validate the variable's value.* See possible responses below:

    a). A 'Y' will allow program to continue to the next step.
    b). A '#' will skip to part b of step 6.
    c). Any other response other than 'Y' or '#' will cause the program to return to step 3.

5. The area represented by #'s is cleared (see SUBROUTINE 2060).
6. a). The variable is now placed in its assigned area. The program continues to step 7.
   b). The #'s are left in the area assigned to the variable under consideration.
7. The size of Y$ is increased by the size, P, of the variable under consideration.
8. The program now moves to the next variable and returns to step 1, until all variables have been viewed.

* Unless otherwise noted a 'Y' response to a validation request will allow the continuation of the program's sequential flow; while any other response will cause the repetition of the question.

PERDAT

720 Conditional branch; the keyboard input of INPUT A and INPUT B is
720 - 1460 directly related to the number of channels to be recorded. The value representing the number of channels obtained here is used in the plotting of the evoked potential data. The user is asked to input: (a) the electrode placement of INPUT A (positive); and (b) the electrode placement of INPUT B (negative). The user starts with the input for the highest numbered channel and ends with the input to channel 1. If there are, for example, only six channels being recorded, the following would happen:

> The markers (#) are left in the ranges of channels 7 and 8, then steps 1, 2, 3, 5, 6a, 7, and 8 of the steps followed in lines 80 - 710 are followed.

1470 After all data for the channels has been entered, a listing of the
470 - 1540 channels and their INPUT A's and INPUT B's is provided and the user is asked to validate all of the variable values. If all values are correct an 'N' response will cause the program to advance to line 1720.

1550 Input the number of the channel that is to be changed and then
1550 - 1710 input the correct INPUT A and INPUT B values. The user is asked to validate the variable's value. The correct value then replaces the old value stored in Y$ (see SUBROUTINE 1970). The program is then returned to line 1470.

1720 The user is asked whether or not data are to be stored. A 'Y'
1720 - 1850 response will be answered with a request for the file's subscript. An 'N' response will cause the program to skip to line 1890. Any other response will cause the question to be repeated. Disk drive #2 is selected, start of DMEM is set to 0, and the size of DMEM is set to 1K. The user is asked to input the file's subscript and validate the variable's value. The file header is made equal to the first digit of the I.D. number. The file base is made equal to the third and fourth digits of the I.D. number. The data contained in Y$ is changed from normal characters into ASCII code and stored on disk.*

* NOTE: IF FILE ALREADY EXISTS, THE ORIGINAL FILE IS DELEATED WITH NO WARNING TO THE USER.

PERDAT

1860 The user is given the option to go straight to the EPDATA program.
1860 - 1880 A 'Y' response will cause the EPDATA program to run. An
'N' response will allow the program to return to sequential flow.
Any other response will cause the question to be repeated.

1890 The user has the option to return to the pointer program (EPPROG)
2890 - 2910 or to exit the program. A 'Y' response will cause the EPPROG
program to run. An 'N' response will cause the program to exit.
Any other response will cause the question to be repeated.

1920 Stop execution of the program. The following are subroutines.

2090 End of the program.

SUBROUTINES FOR PERDAT

Subroutine 1930 (1930 - 1960)

PURPOSE: This subroutine places a series of markers (#) in Y$ in the space allocated to the input variable.

The range of the For/Next loop is the size of Y$, represented by the W, to the size of Y$ when increased by the size of the variable under consideration (W + P - 1). Each Data Memory (DMEM) point in the increased size of Y$ is filled with #'s. The program returns to its sequential flow.

Subroutine 1970 (1970 - 2050)

PURPOSE: This subroutine enables the user to correct any errors made in the input to a channel's INPUT A or INPUT B.

The user inputs the correct INPUT A for the selected channel and is asked to validate the variable's value. The channels new INPUT A replaces the old value in Y$, in the range specified in lines 1810 - 1880.

The user inputs the correct INPUT B for the selected channel and is asked to validate the variable's value. The channels new INPUT B replaces the old value in Y$, in the range specified in lines 1810 - 1880. The program is returned to its sequential flow.

Subroutine 2060 (2060 - 2080)

PURPOSE: This subroutine enables the user to check the value of the variable just requested.

The user is asked to validate the value of the variable which has just been entered. A 'Y' is the response for yes. An 'N' is the response for no and will cause the question to be repeated. A return is the response for a variable with no value. The program is returned to its sequential flow.

VARIABLES USED IN PERDAT

A$	-	represents individual data points during storage
AIRTP$	-	air temperature in room at start of experiment
BASE	-	file base
BLANK$	-	empty variable used to cleanout '#' which mean no response
CH$	-	number of channels to be recorded
COMNT$	-	any comments on experiment or subject
DATE$	-	date of experiment
INA$	-	new value for INPUT A
INA1$	-	INPUT A for channel 1
INA2$	-	INPUT A for channel 2
INA3$	-	INPUT A for channel 3
INA4$	-	INPUT A for channel 4
INA5$	-	INPUT A for channel 5
INA6$	-	INPUT A for channel 6
INA7$	-	INPUT A for channel 7
INA8$	-	INPUT A for channel 8
INB$	-	new value for INPUT B
INB1$	-	INPUT B for channel 1
INB2$	-	INPUT B for channel 2
INB3$	-	INPUT B for channel 3
INB4$	-	INPUT B for channel 4
INB5$	-	INPUT B for channel 5
INB6$	-	INPUT B for channel 6
INB7$	-	INPUT B for channel 7
INB8$	-	INPUT B for channel 8
GFORC$	-	G-Force used in experiment
HEAD	-	file header
HEART$	-	subject's heart rate at the start of experiment
N1	-	start of range for INPUT A
N2	-	end of range for INPUT A
N3	-	start of range for INPUT B
N4	-	end of range for INPUT B
NAME$	-	name of subject
P	-	size of the variable to be added to Y$
RCH$	-	the channel which has a wrong variable
RUNNO$	-	run number for experiment
S	-	loop control variable
SS	-	value of range's for listing INPUT A & INPUT B
SUB	-	file subscript
SUBNO$	-	subject number
SUBTP$	-	subjects temperature at start of experiment
TIMDA$	-	time-of-day for start of experiment
W	-	place marker for Y$
X$	-	reponse to 'Is variable correct' question is a 'Y' for yes, an 'N' for no, or a return for null (empty) variable
XX$	-	temporary retainer for a variable while doing a visual check on it's value
Y$	-	contains all data for storage

EPDATA

PURPOSE: The EPPDATA program (Evoked Potential Data) is used to automatically collect and store parameter values and evoked potential data.

10 This line zeros 4K of memory (DMEM).

20
20 - 60 Variables to be used in this program are dimensioned and some variables have values assigned.

70
70 - 200 The variable for the subject's identification (I.D.) number is zeroed, and the user inputs the correct value. The user is asked to validate the variable's value*. The file base used for the storage of evoked potentials (E.P.) is made equal to the last two numbers of the I.D. number. The file header is made equal to the first digit (letter) of the I.D. number. This letter is either an 'H' for human, or an 'R' for rhesus monkey. The variable for the file subscript is zeroed. The user is asked to input the file subscript's correct value and to validate that value. This is followed by a check to insure that the value is in the range of 0 to 999. If the value is not in this range, the question is repeated. S is made equal to the file subscript and will later be used as a counter for the file subscript.** A value of zero is placed in the post stimulus delay time.

210
210 - 460 The following steps are taken:

1. Input the value for a variable.
2. If the value has to be in some range there is a check to insure that the value is in this range.
3. If the variable is non-string, its value is placed in a string variable for storage.
4. The value of the variable under consideration is placed in its proper storage location in Y$ by the counter variable W.
5. P is set equal to the size of the variable under consideration.
6. W is increased by P.

* Unless otherwise noted, a 'Y' response to a validation request will allow the continuation of the program's sequential flow, while any other response will cause the repetition of the question.

** NOTE: IF A FILE ALREADY EXISTS, THE ORIGINAL FILE IS DELETED WITH NO WARNING TO THE USER.

EPDATA

Each of the following variables are put through the steps (1 - 6) found on page 3.1.

1. Wrist stimulated: WRISTS.
2. Post stimulus delay time: DELAY.
3. Number of stimuli to be presented per second: STSEC.
4. Stimulus period: STPER is calculated by dividing 1 by the number of stimuli to be presented per second. This value is then multiplied by 1000 to be in the same data range as the other values (starts at step 4).
5. Minimum sweep length: SWPSZ (skips all steps). If the sweep length is an improper value an error message is printed, and the user is asked to input a new value for the sweep length (see SUBROUTINE 1510).
6. Size of memory: SZMEM is calculated by dividing the sweep length by .04.
7. Dwell time: DWELL is calculated by dividing the sweep length by the size of memory to be used. The dwell time is now divided by 1000 to be in the same data range as the other values.
8. The dwell time is multiplied by the size of memory to obtain the correct sweep length.
9. Start of memory: STMEM is set to 0 and along with size of memory, dwell time and length of sweep are stored in memory (starts at step 4).

370
370 - 670 The user is given a chance to find the stimulus threshold level for the subject. When the subject's threshold has been found a 'Q' is typed to turn off the stimulus (see SUBROUTINE 1620).

It is now requested that the user input values for the following variables (see steps 1 - 6 found on page 3.1):

1. Subject's threshold: THRES$.
2. Stimulus intensity: STINT$.
3. Amplifier/filter's sensitivity: SENTY.
4. Amplifier/filter's low bandpass: LBAND.
5. Amplifier/filter's high bandpass: HBAND.
6. Number of sweeps to be averaged: SWPS.
7. Number of channels to be collected: CHS.

EPDATA

680
680 - 840 The user is offered a listing of all the variables to be used in the experiment. A 'Y' response will provide a list of all variables with an identification number located on the left hand side. An 'N' response will cause the program to skip to line 850. Any other response will cause the question to be repeated.

850
850 - 1190 After looking at the variable list, the user is asked if the values are correct. A 'Y' response will cause the program to skip to line 1200 of the program. An 'N' response will cause the user to be asked which variable is to be changed. Any other response will cause the question to be repeated. The variable to be changed is identified by the numerical value found on the left hand side. The following variables can not be changed as they are not under user control:

1. dwell time
2. stimulus period
3. start of memory
4. size of memory

It should be noted that changing the following variables' values affects other parameter values:

1. stimulus per second - changes stimulus period.
2. sweep length - changes size of memory and dwell time.

Any changes made on the list of variables will also be made on the file of these parameters (see SUBROUTINE 1750). This section is repeated until there is a 'Y' response to the "are variables correct" question.

1200
1200 - 1240 Disk drive # 2 is selected for storage of files. The letter P (for parameter), represented by the value 208, is placed in the file header. The file's base and subscript are entered to form the file name. The data file is then translated into ASCII code and stored on disk.

EPDATA

1250 The size of memory to be used is zeroed. The stimulus is turned
1250 - 1320 on (see SUBROUTINE 1620). To begin averaging waveforms, the user hits any key. Once the averaging is completed the baseline is zeroed and the stimulus is turned off. The user is asked whether or not the waveforms collected are to be stored on disk. A 'Y' response will cause a D (for data), represented by the value 196, to appear in the file header. The file base and file subscript remain the same as for the parameter file. The data file is then stored on disk #2. The subscript number is now incremented by one, in case the user wants to rerun the program. An 'N' response will cause the program to skip to line 1330, any other response will cause the question to be repeated.

1330 The user is asked whether or not a plot of the evoked potential
1330 - 1400 data is wanted. A 'Y' will cause a plot of all channels to appear on the Hewlett Packard X-Y plotter®. An 'N' response will cause the program to skip to line 1440. Any other response will cause the question to be repeated. The user is now asked whether or not there is to be a second plot of the data enabling a change in the vertical display scale (VDS). A 'Y' response will cause a plot to be produced (make any VDS changes before typing 'Y'). An 'N' response will cause the program go to line 1410, any other response will cause the question to be repeated.

1410 The user is asked if the program is to be rerun without any changes
1410 - 1460 in the parameter values. An 'N' response will cause the program to skip to line 1470. A 'Y' response will, allow the continuation of the program's sequential flow. Any other response will cause cause the question to be repeated. The user is asked if there are any parameter values to be changed. A 'Y' response here will make the program go to line 680; an 'N' response will make the program go to line 1250, any other response will cause the question to be repeated.

1470 The user is asked whether or not the pointer program EPPROG is to
1470 - 1490 be run. A 'Y' response will execute the EPPROG; an 'N' response will make the program go to line 1830, any other response will cause the question to be repeated.

1500 Stop execution of the program. The following are subroutines.

1790 End of program.

SUBROUTINES FOR EPDATA

SUBROUTINE 1510 (1510 - 1610)

PURPOSE: This subroutine is used to calculate the amount of memory to be used, the dwell time, and the length of the sweep to be made in averaging evoked potentials.

To determine the size of memory, the length of the sweep is divided by .04. The size of memory is then set equal to the next largest power of 2. To determine the dwell time, the length of sweep is divided by the size of memory and the remainder is then multiplied by 1000. The dwell time cannot be less than 40 μsec., therefore if the value of the dwell time is less than 40 the time is set equal to 40 μsec. The dwell time is now divided by 1000 to place it in the same millisecond range as the other time parameter value. This new dwell time is multiplied by the size of memory to find the correct sweep length.

SUBROUTINE 1620 (1620 - 1740)

PURPOSE: This subroutine is used to turn on the stimulus, with parameters pre-set in the NIC-293 I/O Controller Unit.

The stimulus is turned on using the following parameters: stimu!us period, stimulus period x 1000, and the delay time. A negative trigger is selected and the user is able to view the ongoing evoked potential signal being received. This subroutine is also used to start the stimulus for finding the subject's threshold and for the collection of averaged evoked potentials (see Appendage 1).

SUBROUTINE 1750 (1750 - 1790)

PURPOSE: This subroutine enables the user to make changes in the values of any variables to be used in the experiment.

The user is asked to input the value of the parameter variable to be changed and then validates the response. The new parameter value is placed in its correct position in the parameter file.

VARIABLES USED IN EPDATA

A$	-	represents individual data points from Y$ during storage
CHS	-	number of channels being recorded
CHS$	-	number of channels being recorded
DELAY	-	delay time before recording
DELAY$	-	delay time before recording
DWELL	-	dwell time
DWELL$	-	dwell time
HBAND	-	high bandpass
HBAND$	-	high bandpass
LBAND	-	low bandpass
LBAND$	-	low bandpass
LLEWD	-	dwell time divided by 1000
N1	-	start of range for the variable being changed
N2	-	end of range for the variable being changed
NO	-	the number of the variable being changed
REPTS	-	stimulus period multiplied by 1000
REPTS$	-	stimulus period multiplied by 1000
S	-	loop control variable
SUB	-	file subscript
SSUB	-	file subscript
SENTY	-	amplifier sensitivity
SENTY$	-	amplifier sensitivity
STINT$	-	stimulus intensity
STMEM$	-	start of memory
STPER	-	stimulus period
STPER$	-	stimulus period
STSEC	-	number of stimuli presented per second
STSEC$	-	number of stimuli presented per second
SUBNO$	-	subject's I.D. number
SWPS	-	number of sweeps to be averaged
SWPS$	-	number of sweeps to be averaged
SWPSZ	-	length of sweep
SWPSZ$	-	length of sweep
SZMEM	-	size of memory
SZMEM$	-	size of memory
THRES$	-	subject's stimulus threshold
VAL$	-	new value for the variable being changed
W	-	place marker for Y$
WRIST$	-	wrist being stimulated
X$	-	response to a YES or NO question
Y$	-	contains all variable data for this program

PLOTEP

PURPOSE: The PLOTEP program (Plot Evoked Potential) is used to plot both evoked potential parameter and waveform data, with the corresponding personal data on the Nicolet Zeta Plotter®.

10 All variables to be used in this program are dimensioned, and
10 - 110 values are assigned to some of the variables.

120 This line sets the starting point of Data Memory (DMEM) to 0 and memory size to 1K. Disk drive #2 which contains the data files is selected.

130 The user is asked to input the subject's identification (I.D.) num-
130 - 310 ber, and then validate the response.* The file base is made equal to the last two digits of the I.D. number, while the file header is made equal to the first digit (a letter) of the I.D. number. The user inputs the subscript number, which is to be used in retrieving data files, and validates the response. This is followed by a check to insure that the value is in the correct data range. A conditional branch is used on the value of the file header to see whether the file header represents a human file (read value of 40 or H), or a rhesus file (read value of 50 or R). The correct file header (write values of 200 = H or 210 = R), file base and file subscripts are stored in DMEM. The selected personal file is read from disk and translated from ASCII code into normal characters. These characters are then stored in V$. A P represented by 208 is put into the file header in order to call up the parameter file. The file is read from disk using the file base and subscript used in calling the personal file. The new file is translated from ASCII code into normal characters, and placed in Y$.

320 The sweep length, HS, and the number of channels recorded, O$, are obtained from V$. CH is made equal to the string variable for the number of channels recorded, CH$. The increments to be used on the horizontal time scale for the sweep length are determined by dividing the sweep length by 10.

* Unless otherwise noted a 'Y' response to a validation request will allow the continuation of the program's sequential flow, while any other response will cause the repetition of the question.

PLOTEP

330 330 - 410 The user inputs the minimum voltage level, VA, obtained from the experiment and validates the variable's value. VR is made equal to the minimum voltage level times 2 to get the full (+/-) range. This number is divided by ten, giving the size of the increments to be used on the vertical voltage scale. The user inputs the gain which is to be used in the plotting of the evoked potential (E.P.) and validates the variable's value.

420 The Zeta Plotter's® pen should be placed in the upper right hand corner when facing the plotter. The current pen position of the plotter is selected to be the point of reference (origin). The rotation of the pen is selected as 1.0.*

```
                  ________________
                  |              |
                  |              |
          )-->    |              |
     USER         |              |   <------------------
          )-->    |              |     flow of paper
                  |              |
                  |_____________x|   x - current position of pen
                                             called 0.0
```

430 SUB$ is made equal to the SUB. A variable which represents the filename of the E.P. file to be plotted 'NOF$', is built by having its first digit as the first digit of the I.D. number, its second and third digits as the third and forth digits of the I.D. number, and its fourth, fifth, and sixth digits as the file subscript number.

450 450 - 480 Via a conditional branch, the program advances to the location which contains the correct format for plotting the number of channels recorded.

* See Zeta Plotter® - SBASIC Commands handout M0290601 for SUP800.

PLOTEP

490
490 - 1170 The following are steps to be followed in the plotting of experimental data:

1. Plotting paper is advanced and the point of origin for the Zeta Plotter® is selected.
2. Value is given to the variable which determines the distance that channel information is to be printed from the bottom of the page.
3. The number of channels to be plotted is obtained from the personal file.
4. The montage information for the selected channel is obtained from the parameter file.
5. The channel number and the montage information is plotted in its proper location.
6. The top vertical line for the box containing the bottom E.P. waveform to be plotted is drawn with incremented tick marks.
7. The header information for the vertical scale is drawn at a right angle (2.0 rotation) to the other header information.
8. The rest of the header information is printed at the top of the page. This is repeated until the container and header information for all E.P. channels has been plotted.
9. The value for the size of DMEM, SZ, is obtained from the parameter file. The size of memory for each channel, PLTSZ, is determined (PLTSZ = SZ / CH). The size of DMEM for the E.P. file is input into BASEXC. Disk drive #2 is selected. The value 196, representing 'D' for E.P. data file, is put into the file header. The file base number and the subscript number used to select the personal and parameter files are used to select the E.P. data file. The E.P. data file is then read from disk. The size of memory used in storing each individual channel is placed in DMEM. The present pen location is made the point of origin, and a rotation of 1.0 is selected.
10. A conditional branch for the plotting of E.P. waveforms is dependent on the number of channels which were recorded during the experiment.
11. The graph paper is advanced to the page where the highest numbered channel is to be plotted and the point of origin is found. The E.P. waveform for that channel is read from disk and plotted, followed by the plotting of all other waveforms waveforms in their prospective boxes.
12. After the plotting of channel one's waveform the pen is returned to the point of origin.

PLOTEP

1180 The user is asked whether or not another file is to be plotted. If
1180 - 1250 the response is 'Y' then the program continues with its sequential flow. An 'N' response will advance the program to line 1310. Any other response will cause the question to be repeated. The user is asked whether or not the next file to be plotted is in the same series as the previous file. A 'Y' response will advance the program to line 1230. If the response is 'N' the program advances graph paper to a clean page (see SUBROUTINE 2100) and the program is returned to line 120. Any other response will cause the question to be repeated. The user is asked whether or not the next file to be plotted is the next number in the series. A 'Y' response will increase the file subscript by one, advance the graph paper to a clean page (see SUBROUTINE 2100), and return the program to line 420. An 'N' response will advance the program to line 1260, any other response will cause the question to be repeated.

1260 Input the subscript of the new file and validate the value. A 'Y'
1260 - 1300 response will advance the graph paper (see SUBROUTINE 2100) and return the program to line 420, while any other response will cause the question to be repeated.

1310 The user is asked whether or not another program in the E.P. series
1310 - 1330 is to be run. A 'Y' response will initiate the Pointer Program, while an 'N' response will cause the program to exit. Any other response will cause the program to repeat the original question.

1335 Stop execution of the program. The following are subroutines.

2190 End of program.

SUBROUTINES FOR PLOTEP

SUBROUTINE 1340 (1340 - 1380)

PURPOSE: The purpose of this subroutine is to plot a channel's montage information in the correct location on the page.

The number of the channel being plotted is printed. The channel's correct INPUT A montage is pulled off of the parameter file and printed. A dash is then printed to separate the montages. The channel's correct INPUT B montage is pulled off of the parameter file and printed. The program then returns to its sequential flow.

SUBROUTINE 1390 (1390 - 1420)

PURPOSE: The purpose of this subroutine is to draw the header information for the vertical voltage scale at a right angle to all other header information.

The Zeta Plotter® pen is returned to its point of origin. The rotation of the pen direction is changed to 2.0, and the pen is moved into position for plotting. The header information for the vertical voltage scale is printed. The pen is moved to the bottom of the preceding page. This location is made the new point of origin with a rotation of 1.0. The program is returned to its sequential flow.

SUBROUTINE 1430 (1430 - 1460)

PURPOSE: The purpose of this subroutine is to draw the top of the enclosure which contains the E.P. waveform for a file with an odd number of recorded channels. Only the lower box is drawn for the individual channel.

The value of XX is set to -10. The Zeta Plotter® pen is moved into position and the top horizontal line is drawn for the E.P. waveform box. The eleven incremented tick marks on the top horizontal line. All header information and the rest of the box is now printed (see SUBROUTINE 1470).

SUBROUTINES FOR PLOTEP

SUBROUTINE 1470 (1470 - 1920)

PURPOSE: The purpose of this subroutine is to plot the header information and the rest of the bottom enclosure for plotting an E.P. waveform.

All header information is plotted. The bottom horizontal line with incremented tick marks and the time scale is plotted, along with its header information. The left vertical line is plotted with it's incremented tick marks on the voltage scale. The tick marks for the center line of the lower box are plotted followed by the plotting of the lower right vertical line with incremented tick marks.

SUBROUTINE 1930 (1930 - 2090)

PURPOSE: The purpose of this subroutine is to plot the enclosures for the upper E.P. enclosure.

The upper left vertical line with the voltage scale is drawn with incremented tick marks. The top horizontal line is plotted with incremented tick marks, as is the right vertical line. The center line for the upper enclosure is now plotted. The program is returned to its sequential flow.

SUBROUTINE 2100 (2100 - 2180)

PURPOSE: The purpose of this subroutine is to advance the graph paper to a clean page for the plotting of another E.P. file.

The paper is advanced via a conditional branch according to the number of channels plotted in the last execution of the program. The program is returned to its sequential flow.

VARIABLES USED IN PLOTEP

A$ - G-FORCE
B$ - DATE:
BASE - file base number
C - distance incrementation for tick marks
CH - number of channels recorded
CH$ - number of channels being plotted
C$ - NAME:
C1$ - CH.#1
C2$ - CH.#2
C3$ - CH.#3
C4$ - CH.#4
C5$ - CH.#5
C6$ - CH.#6
C7$ - CH.#7
C8$ - CH.#8
D - distance from bottom of page for plotting channel information
D$ - SUBJECT NUMBER:
E$ - SLED RUN NUMBER:
F$ - SOMATOSENSORY EVOKED POTENTIAL
FN$ - FILENAME:
G$ - (IN MSEC.)
GA$ - PLOT GAIN:
GAIN$ - gain to be used in plotting of E.P. waveforms
H$ - (IN UV)
HEAD - file header
HR - incrementation for length of sweep scale
HS - length of sweep
I - loop control variable
I$ - DELAY TIME:
J$ - STIMULI/SECOND:
K$ - DWELL TIME (IN US):

VARIABLES USED IN PLOTEP

L$	- SWEEP LENGTH:
M	- distance incrementation for tick marks
M$	- -
N1	- start of location for 'INPUT A' montage
N2	- end of location for 'INPUT A' montage
N3	- start of location for 'INPUT B' montage
N4	- end of location for 'INPUT B' montage
NO$	- NUMBER SWEEPS:
NOF$	- name of data f,ie tc be plotted
O$	- number of channels recorded
P	- distance for incrementation of dashes
PP	- individual data points from DMEM to Y$
PLTSZ	- size of memory used in storage of individual waveforms
SUB	- file subscript number
SUB$	- file subscript number
SUBNO$	- subject's I.D. number
SZ	- size of memory used in the storage of waveforms
U	- ASC data from data memory for V$
UU	- individual data points from DMEM to V$
UHA	- distance for plotting length of sweep scales
UVA	- distance for plotting voltage level scale
V	- ASC data from data memory for Y$
VA	- voltage level in E.P. recording
VR	- incremenation for the range of the voltage level
V$	- parameter file
W	- loop control variable
W$	- WRIST STIMULATED:
XX	- distance from edge of paper for making tick marks
X$	- response to a 'Y' or an 'N' question
Y$	- personal data file
Z	- distance from edge of paper for making dashes

LSTPER

PURPOSE: The LSTPER program (List Data) is used to provide a detailed listing of the data contained in the personal data file and the parameter data file.

10 All the variables to be used in this program are dimensioned.

20 The range of data memory (DMEM) is provided with 0 as the start and 1K as the size. Disk drive #2 is selected, as it is here that the data files are stored.

30 The user inputs the subject's identification number (I.D.), and
30 - 140 then validates the value. The file header is the first digit of the I.D. number (a letter), and the file base is the third and fourth digit of the I.D. number. The user now inputs a three digit file subscript to be used in calling the correct data files and validates the subscript's value. The value of the subscript is checked to ensure that it is in the correct data range.

150 A letter H (200) for human, or an R (210) for rhesus is placed in
150 - 200 the file header 'PFH', the file base is placed in 'PFB', and the file subcript is placed in 'PFS'. After the filename has been selected the data file is read from disk into DMEM. Individual data points are read from DMEM, translated from ASCII code into normal characters, and placed in Y$.

210 Input the letter P (208) for parameter in the file header. The
210 - 250 file is read from disk into DMEM. The individual data points are read from DMEM, translated from ASCII code to normal characters, and placed in V$.

260 To continue the program, the user is asked to advance the paper
260 - 280 to the top of a new page and type 'Y'. This allows a separation of unwanted instructions from the desired listing of the personal and parameter data values.

290 The variable in question is printed out followed by its assigned
290 - 710 value. The program advances to the next variable.

LSTPER

720 The user is asked whether or not another personal data file is to
720 - 750 be listed. A 'Y' response will return the program to line 30.
An 'N' response will continue the program in its sequential flow.
Any other response will cause the question to be repeated.

760 The user is asked whether or not another program is to be executed.
760 - 780 A 'Y' response will cause the execution of the pointer program
(EPPROG). An 'N' response will stop the program. Any other response will cause the question to be repeated.

790 End of program.

VARIABLES USED IN LSTPER

BASE	-	file base
HEAD	-	file header
S	-	loop control variable
SUB	-	file subscript
SUBNO$	-	subject's identification number
X	-	ASC data from data memory
X$	-	response to a yes or no question
Y$	-	contains variable values from the personal data file
V$	-	contains variable values from the parameter data file

APPENDIX 1

PATCH PANEL CONNECTIONS MADE ON THE NIC-293®

PATCH PANEL CONNECTIONS MADE ON THE NIC-293® *

SOURCE	DESTINATION
B-C2	B-D7
B-C1	A-A1
B-C6	B-D5
B-C4	A-C8
A-C8	A-J1
B-C2	A-A3
A-J2	B-D3
B-E5	B-D4
A-G3	B-D6
B-H3	B-J1
B-J1	B-J2
B-J2	B-J3

FUNCTIONS OF PINS **

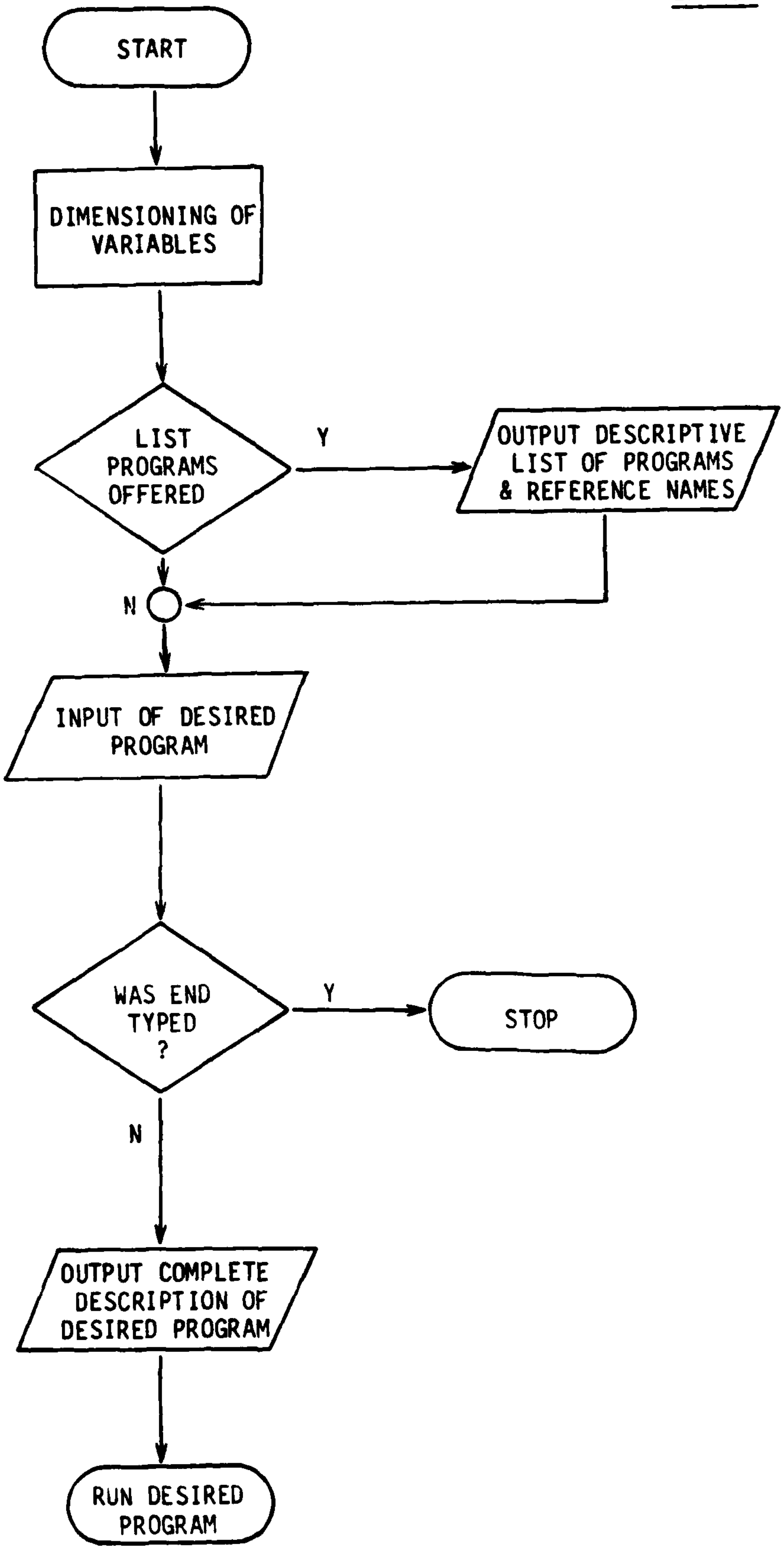
A-A1 - BNC connector A
A-A3 - BNC connector C
A-C8 - Amphenol terminal V
A-G3 - I/O level output (if the I/O level is low then the timer is enabled; if I/O level is high then the timer is disabled)
A-J1 - One Shot input (0 to 4 volt transition)
A-J2 - One Shot output (100 nanosecond pulse)
B-C1 - (-)1 Buffered timing pulse output
B-C2 - (+)1 Buffered timing pulse output
B-C4 - (-)2 Buffered timing pulse output
B-C6 - (-)3 Buffered timing pulse output
B-D3 - ORed and buffered trigger inputs for trigger A1
B-D4 - ORed and buffered trigger inputs for trigger B1
B-D5 - ORed and buffered trigger inputs for trigger A2
B-D6 - ORed and buffered trigger inputs for trigger B2
B-D7 - ORed and buffered trigger inputs for trigger A3
B-E5 - unassigned decoded I/O command used to start timers
B-H3 - Internal clock frequency (5 MHz)
B-J1 - Timer clock # 1
B-J2 - Timer clock # 2
B-J3 - Timer clock # 3

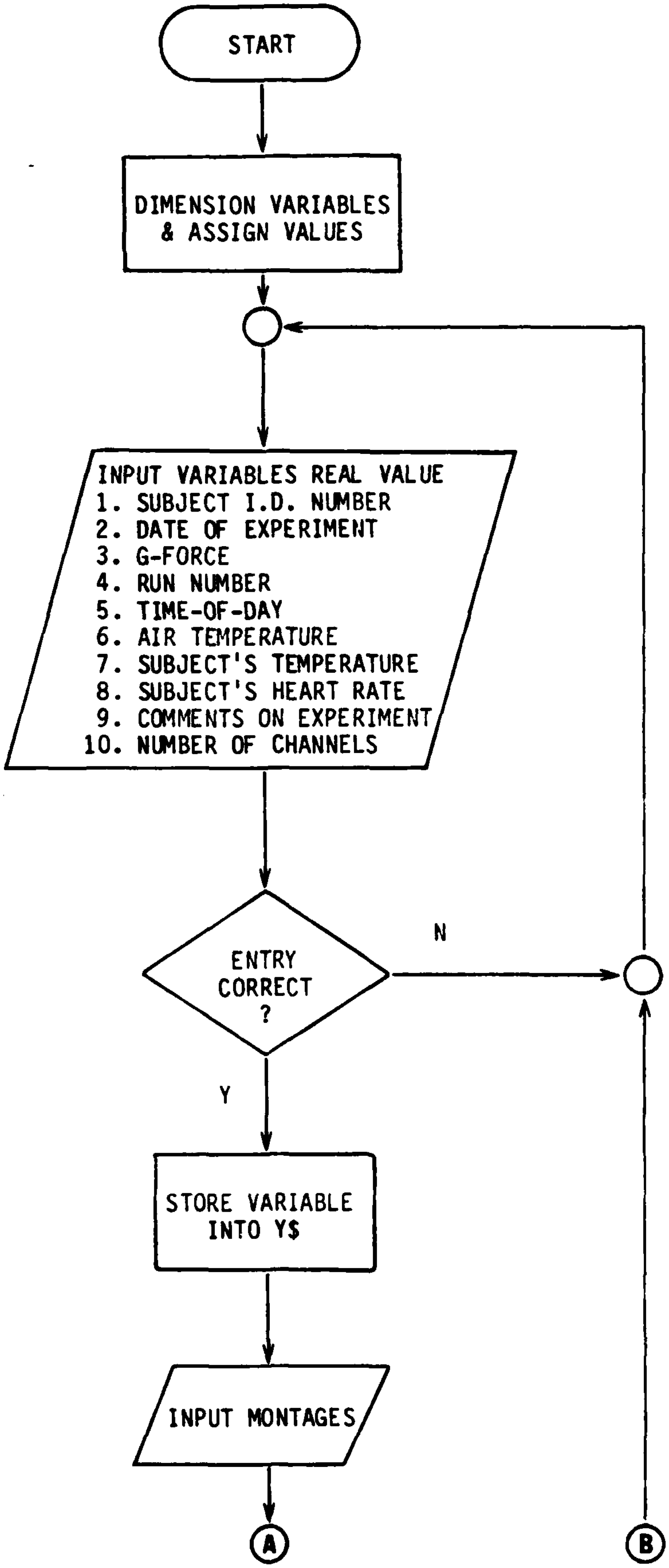
* See NIC-293®Timer Configuration Handout for Prestimulus Averaging.

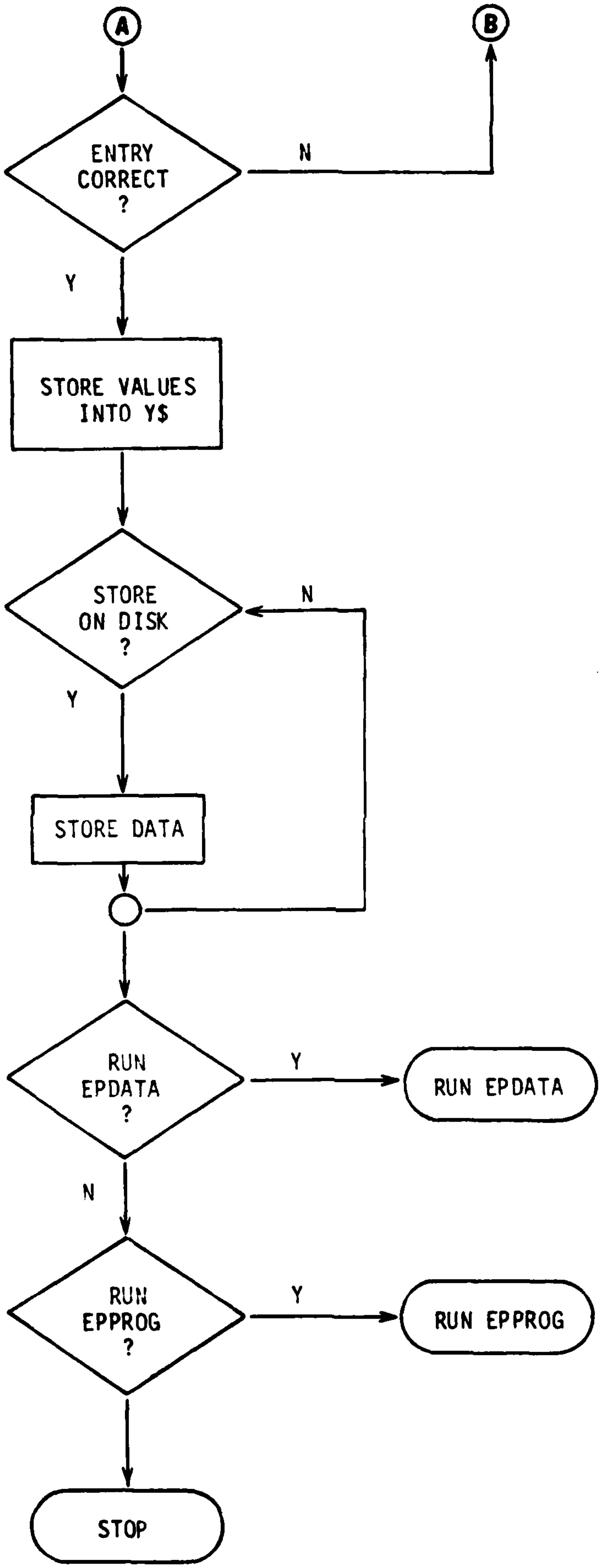
** See NIC 1080® Series Description and Instructions Manual for NIC-293 I/O Controller®.

APPENDIX 2

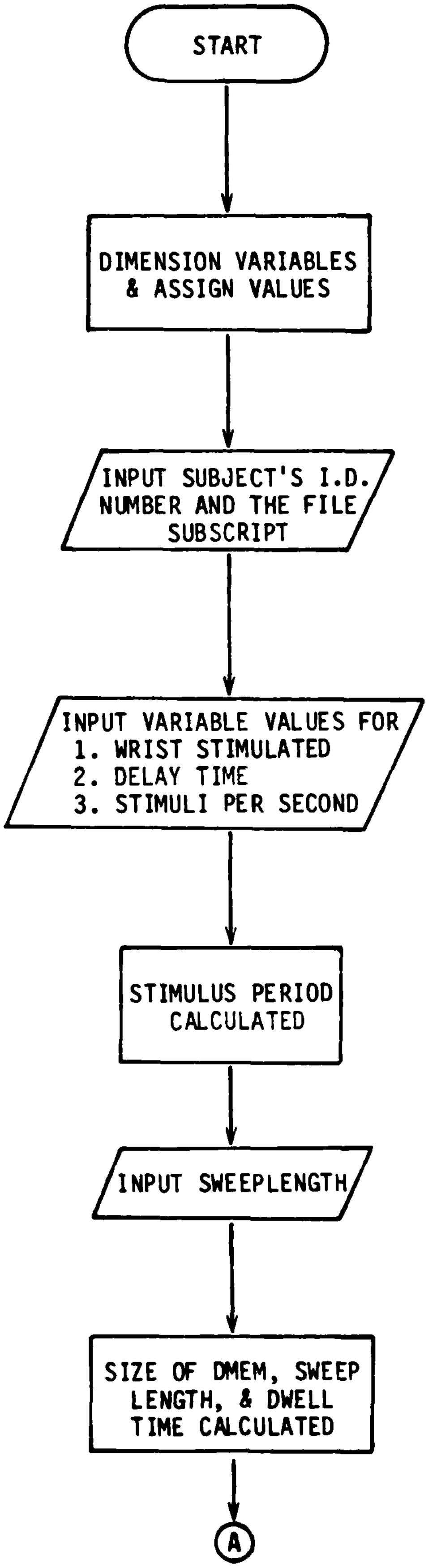
FLOWCHARTS

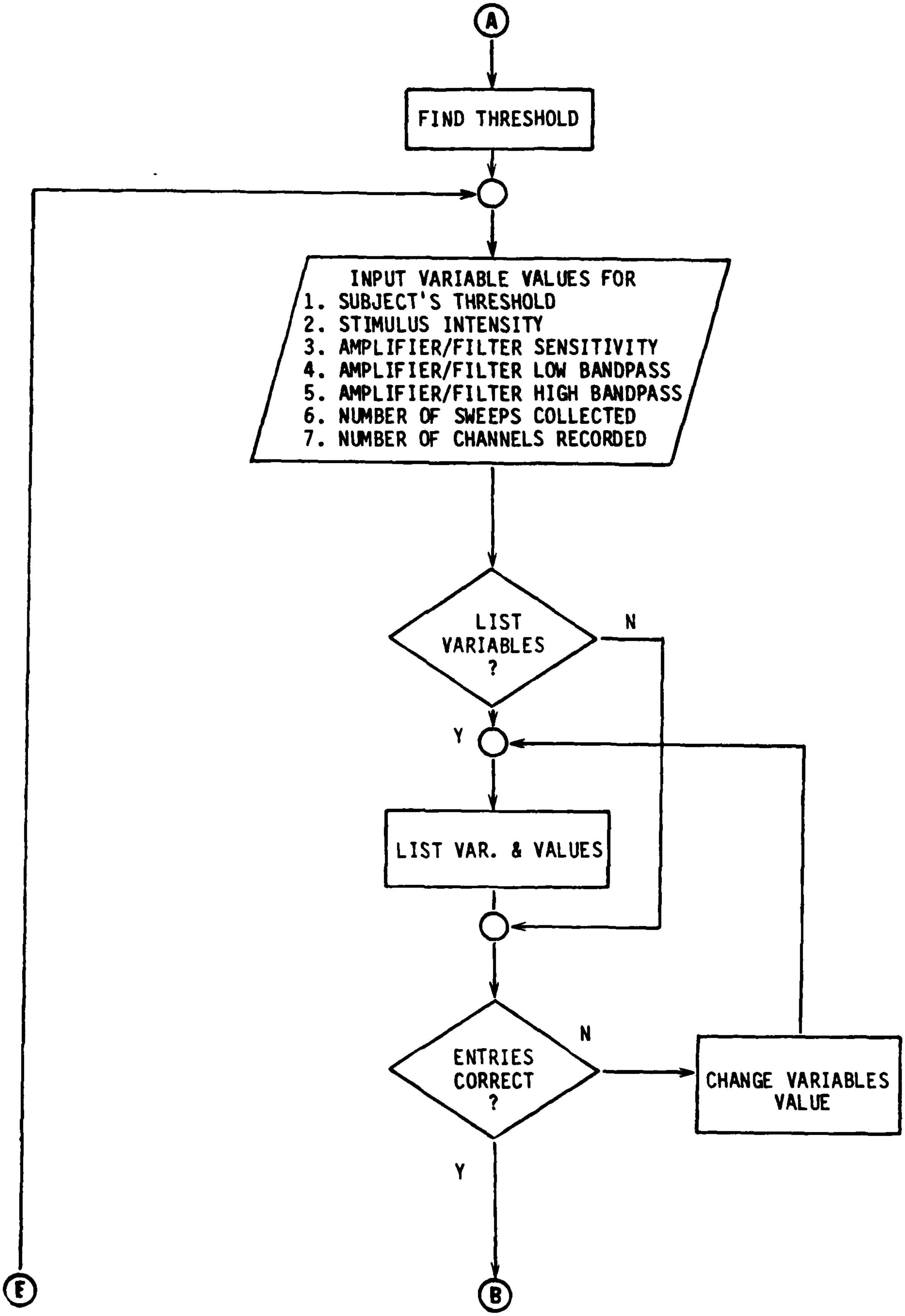

START
DIMENSIONING OF VARIABLES
LIST PROGRAMS OFFERED
Y
OUTPUT DESCRIPTIVE LIST OF PROGRAMS & REFERENCE NAMES
N
INPUT OF DESIRED PROGRAM
WAS END TYPED ?
Y
STOP
N
OUTPUT COMPLETE DESCRIPTION OF DESIRED PROGRAM
RUN DESIRED PROGRAM

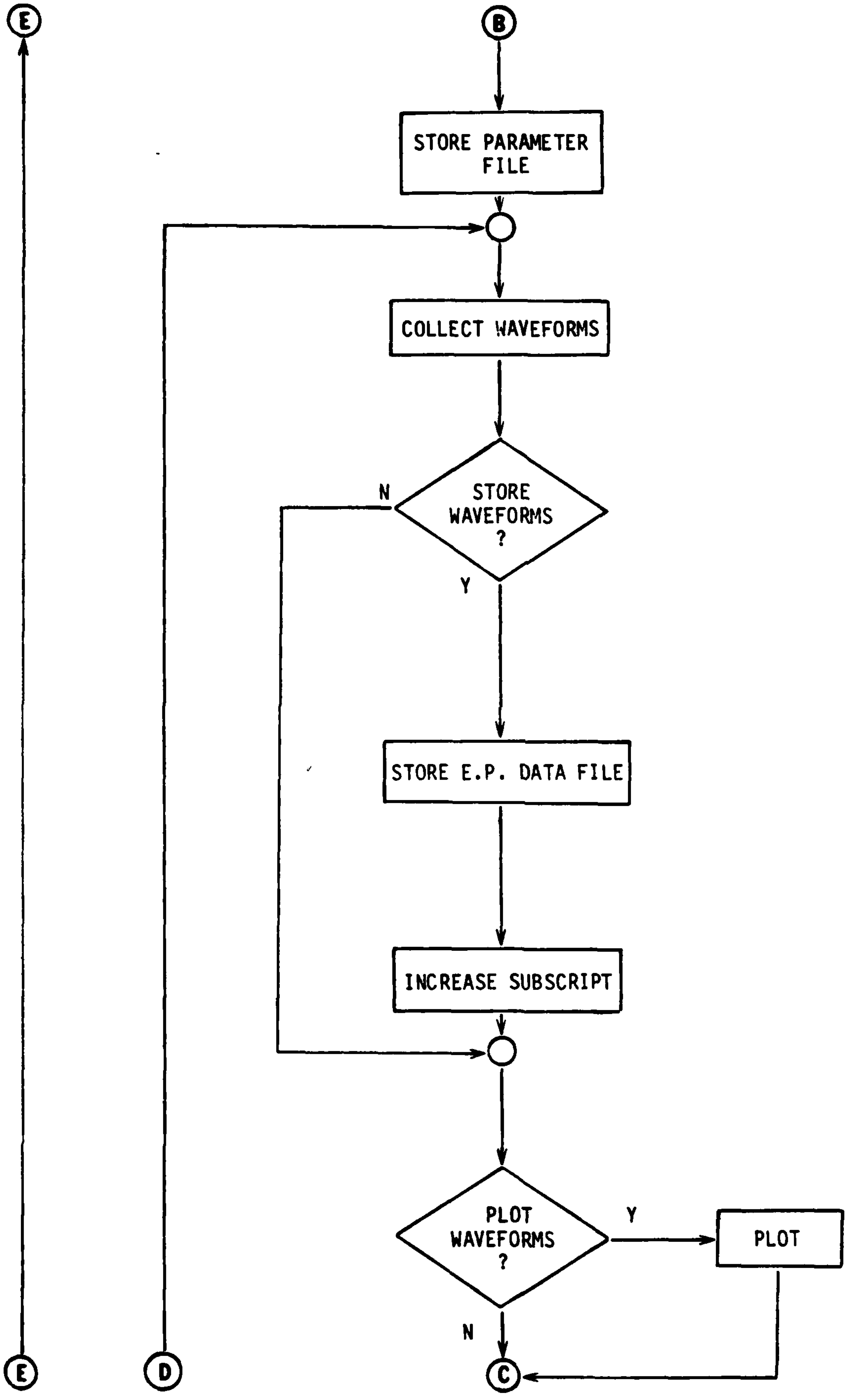

START
DIMENSION VARIABLES & ASSIGN VALUES
INPUT VARIABLES REAL VALUE
1. SUBJECT I.D. NUMBER
2. DATE OF EXPERIMENT
3. G-FORCE
4. RUN NUMBER
5. TIME-OF-DAY
6. AIR TEMPERATURE
7. SUBJECT'S TEMPERATURE
8. SUBJECT'S HEART RATE
9. COMMENTS ON EXPERIMENT
10. NUMBER OF CHANNELS
ENTRY CORRECT ?
N
Y
STORE VARIABLE INTO Y$
INPUT MONTAGES
A
B

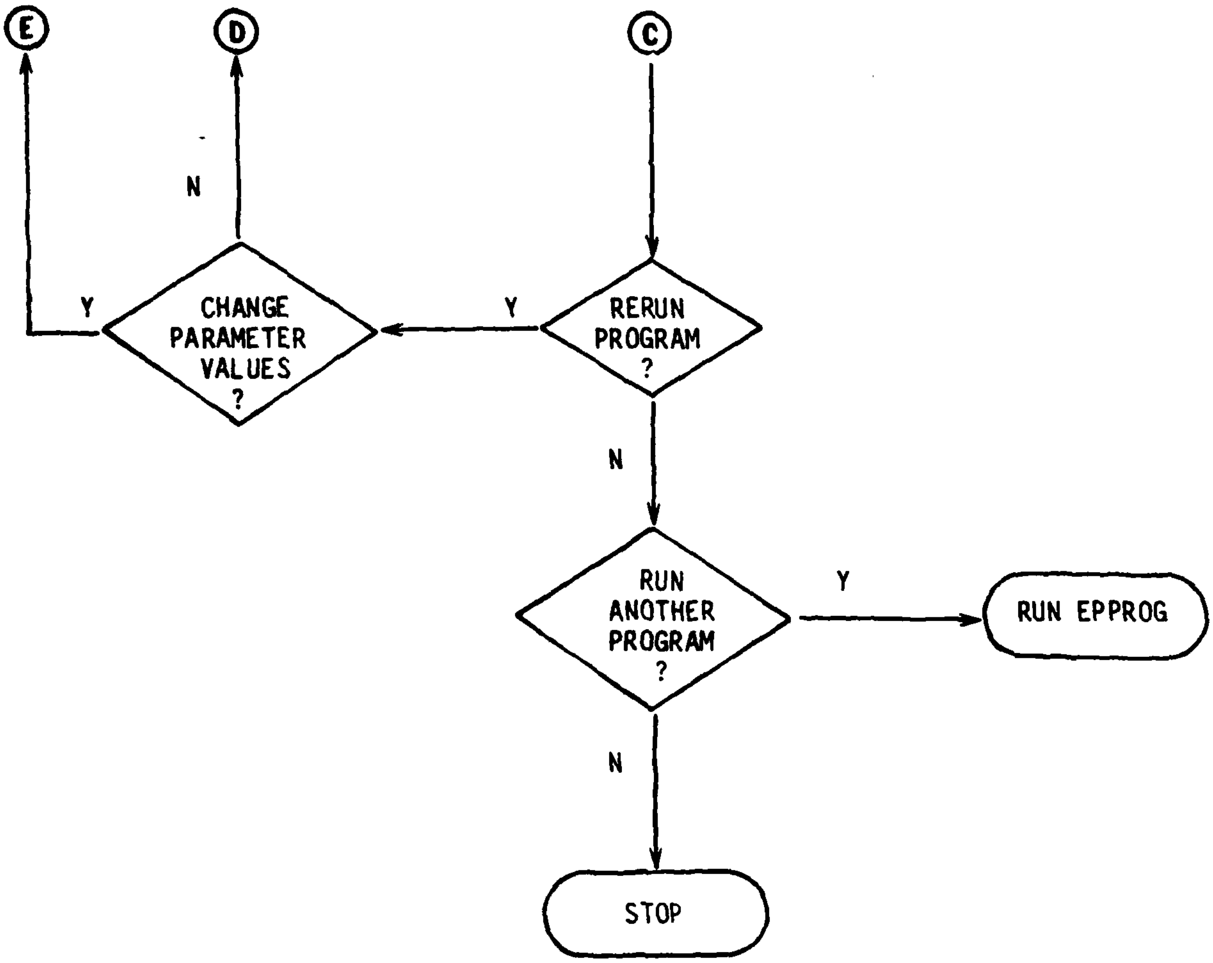

A
B
ENTRY CORRECT ?
N
Y
STORE VALUES INTO Y$
STORE ON DISK ?
N
Y
STORE DATA
RUN EPDATA ?
Y
RUN EPDATA
N
RUN EPPROG ?
Y
RUN EPPROG
STOP

EPDATA

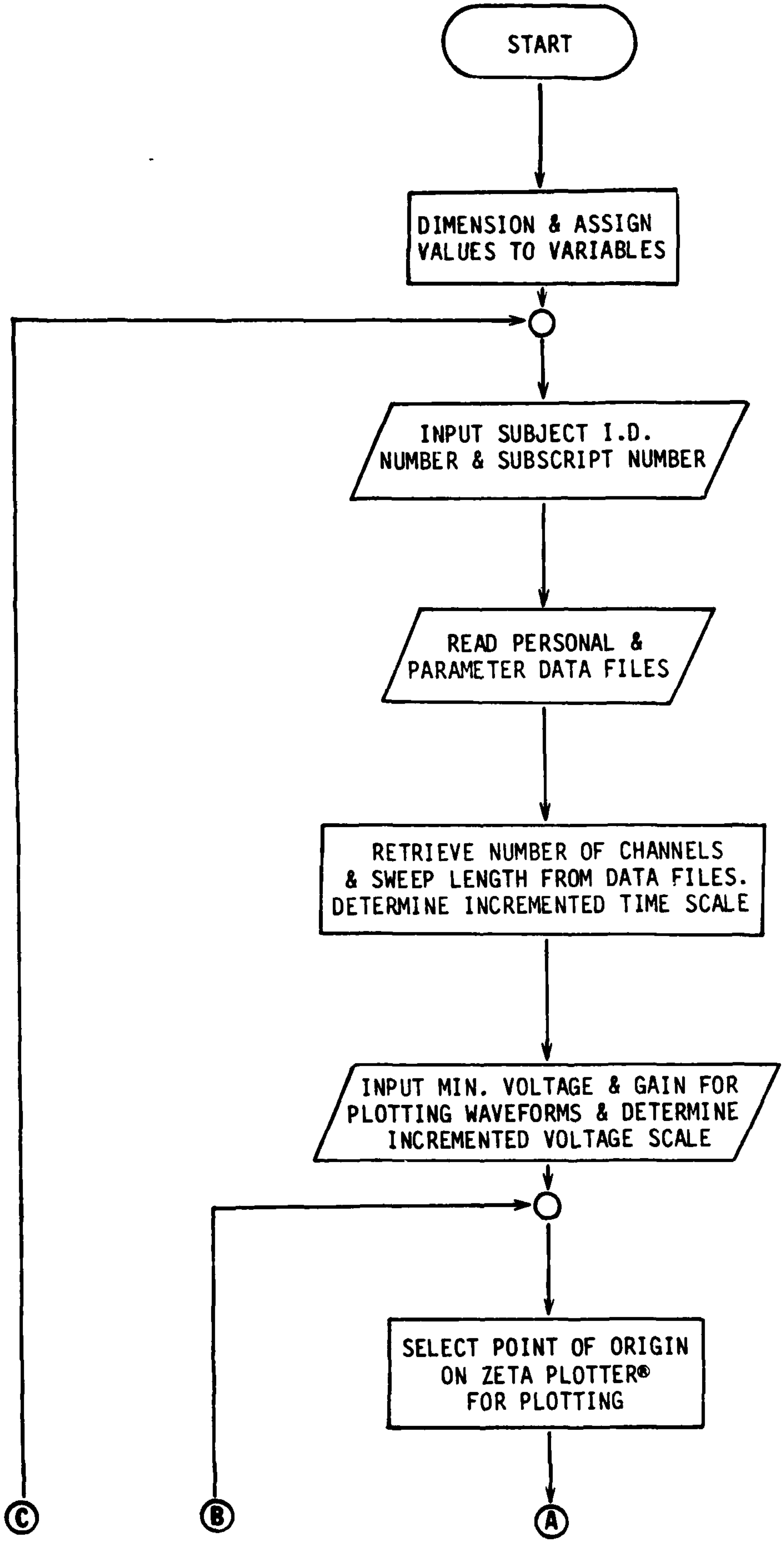

START
DIMENSION VARIABLES & ASSIGN VALUES
INPUT SUBJECT'S I.D. NUMBER AND THE FILE SUBSCRIPT
INPUT VARIABLE VALUES FOR
1. WRIST STIMULATED
2. DELAY TIME
3. STIMULI PER SECOND
STIMULUS PERIOD CALCULATED
INPUT SWEEPLENGTH
SIZE OF DMEM, SWEEP LENGTH, & DWELL TIME CALCULATED
A

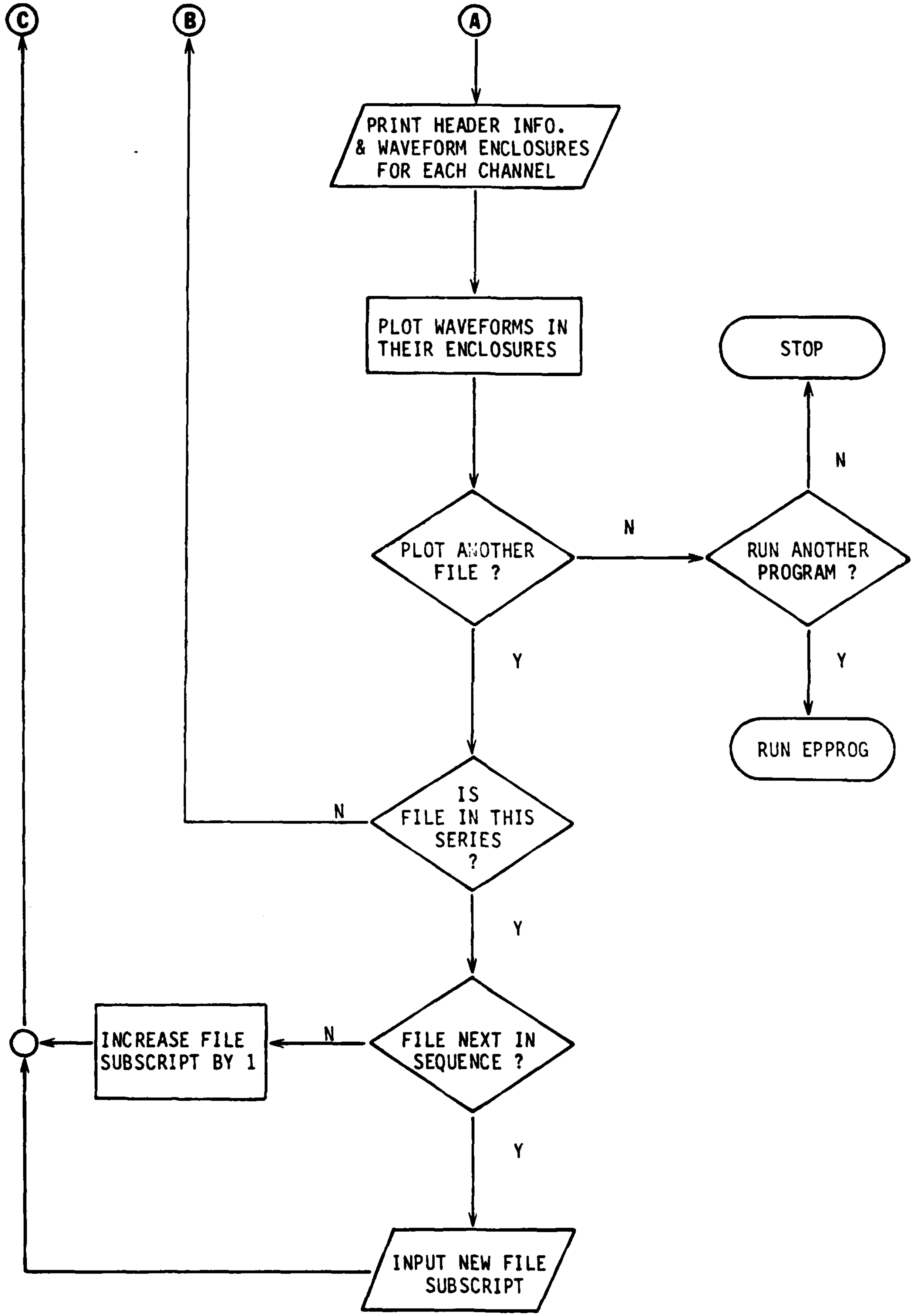

A
FIND THRESHOLD
INPUT VARIABLE VALUES FOR
1. SUBJECT'S THRESHOLD
2. STIMULUS INTENSITY
3. AMPLIFIER/FILTER SENSITIVITY
4. AMPLIFIER/FILTER LOW BANDPASS
5. AMPLIFIER/FILTER HIGH BANDPASS
6. NUMBER OF SWEEPS COLLECTED
7. NUMBER OF CHANNELS RECORDED
LIST VARIABLES ?
N
Y
LIST VAR. & VALUES
ENTRIES CORRECT ?
N
CHANGE VARIABLES VALUE
Y
E
B


E
B
STORE PARAMETER FILE
COLLECT WAVEFORMS
N
STORE WAVEFORMS ?
Y
STORE E.P. DATA FILE
INCREASE SUBSCRIPT
PLOT WAVEFORMS ?
Y
PLOT
N
E
D
C

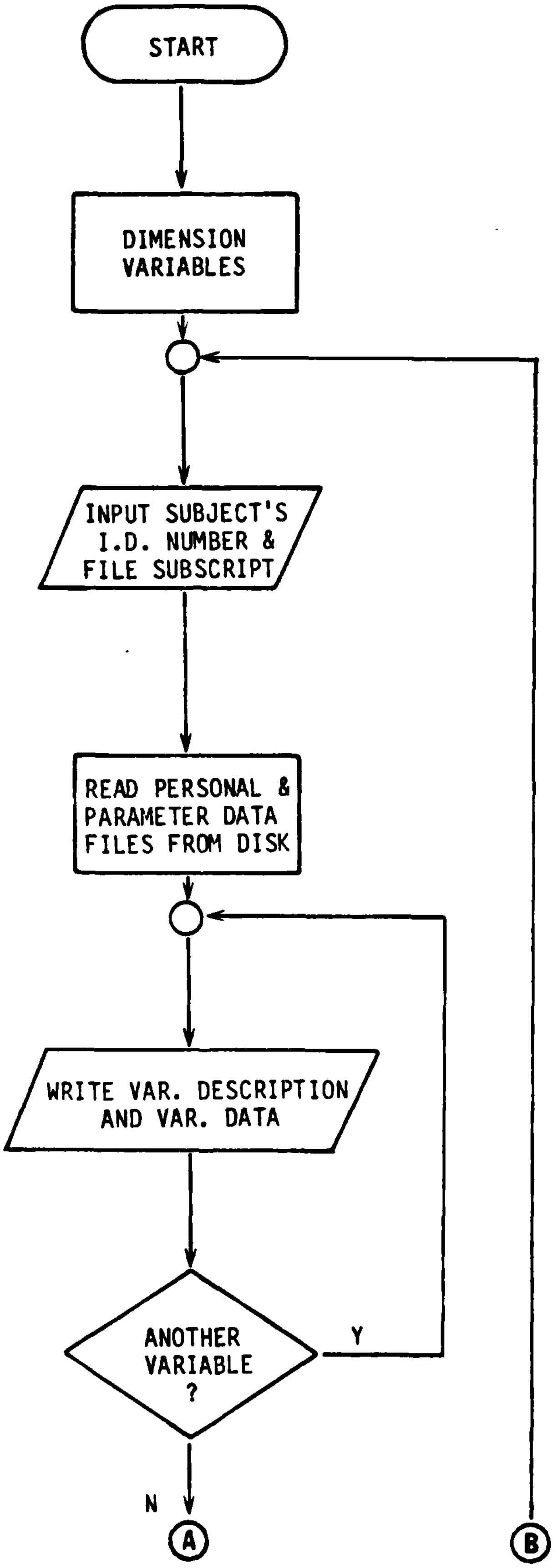

E
D
C
N
Y
CHANGE PARAMETER VALUES ?
Y
RERUN PROGRAM ?
N
RUN ANOTHER PROGRAM ?
Y
RUN EPPROG
N
STOP

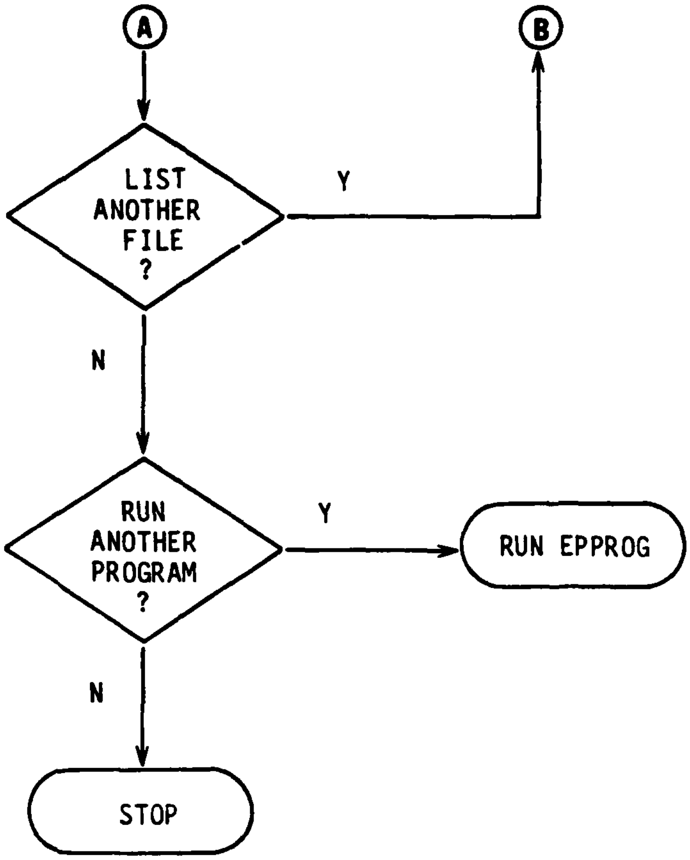
PLOTEP


START
DIMENSION & ASSIGN VALUES TO VARIABLES
INPUT SUBJECT I.D. NUMBER & SUBSCRIPT NUMBER
READ PERSONAL & PARAMETER DATA FILES
RETRIEVE NUMBER OF CHANNELS & SWEEP LENGTH FROM DATA FILES. DETERMINE INCREMENTED TIME SCALE
INPUT MIN. VOLTAGE & GAIN FOR PLOTTING WAVEFORMS & DETERMINE INCREMENTED VOLTAGE SCALE
SELECT POINT OF ORIGIN ON ZETA PLOTTER® FOR PLOTTING
C
B
A


C
B
A
PRINT HEADER INFO.
& WAVEFORM ENCLOSURES
FOR EACH CHANNEL
PLOT WAVEFORMS IN
THEIR ENCLOSURES
STOP
N
PLOT ANOTHER
FILE ?
N
RUN ANOTHER
PROGRAM ?
Y
Y
RUN EPPROG
IS
FILE IN THIS
SERIES
?
N
Y
INCREASE FILE
SUBSCRIPT BY 1
N
FILE NEXT IN
SEQUENCE ?
Y
INPUT NEW FILE
SUBSCRIPT

LSTPER


START
DIMENSION VARIABLES
INPUT SUBJECT'S I.D. NUMBER & FILE SUBSCRIPT
READ PERSONAL & PARAMETER DATA FILES FROM DISK
WRITE VAR. DESCRIPTION AND VAR. DATA
ANOTHER VARIABLE ?
Y
N
A
B


A
B
LIST ANOTHER FILE ?
Y
N
RUN ANOTHER PROGRAM ?
Y
RUN EPPROG
N
STOP

APPENDIX 3

PRINTOUTS OF EACH PROGRAM

```
LIST EPPROG
$
   10  DIM PROG$(6) , X$(1)
   20  PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT
   30  PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT
   40  PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT
   50  PRINT "DO YOU NEED A LISTING OF THE PROGRAMS BEING OFFERED OR"
   60  INPUT "DO YOU NEED SOME INFORMATION ON THE PURPOSE OF THE PROGRAMS? - ", X$
   70  IF X$ = "Y" THEN 80 ; IF X$ = "N" THEN 230 ; GOTO 50
   80  PRINT ; PRINT "    THE PROGRAM DISK SHOULD BE IN DISK DRIVE # 1."
   90  PRINT "          THE DATA DISK SHOULD BE IN DISK DRIVE # 2."
  100  PRINT ; PRINT " NOTE:  IF YOU MAKE A ERROR NO ERROR MESSAGE IS PRINTED."
  110  PRINT "       THE QUESTION WILL JUST BE REASKED."
  120  PRINT ; PRINT
  130  PRINT " THE FOLLOWING IS A LISTING OF ALL PROGRAMS IN THE EVOKED POTENTIAL"
  140  PRINT "   SERIES AND THE SYMBOLS TO BE USED WHEN CALLING A PROGRAM."
  150  PRINT ; PRINT "INPUT OF PERSONAL DATA - PERDAT"
  160  PRINT "COLLECTION OF PARAMETER AND EVOKED POTENTIAL DATA - EPDATA"
  170  PRINT "PLOTTING OF THE EVOKED POTENTIAL DATA ON THE ZETA PLOTTER - PLOTEP"
  180  PRINT "OBTAINING A DETAILED LISTING OF ALL EXPERIMENTAL DATA - LSTPER"
  190  PRINT "TRANSFER OF DATA FILES TO THE ECLIPSE - MEDECL"
  210  PRINT ; PRINT "IF YOU DESIRE TO EXIT THE PROGRAM TYPE 'END' AS A "
  220  PRINT "   RESPONSE TO THE FOLLOWING QUESTION."
  230  PRINT ; INPUT "INPUT THE SYMBOL OF THE DESIRED PROGRAM - ", PROG$(1)
  240  PRINT "IS ", PROG$(1), " CORRECT?"
  250  INPUT X$(1)
  260  IF X$(1) = "Y" THEN 270 ; GOTO 230
  270  PRINT
  280  IF PROG$(1) = "VARUSE" THEN 320 ; IF PROG$(1) = "PERDAT" THEN 350
  290  IF PROG$(1) = "LSTPER" THEN 560 ; IF PROG$(1) = "EPDATA" THEN 600
  300  IF PROG$(1) = "MEDECL" THEN 700 ; IF PROG$(1) = "PLOTEP" THEN 730
  310  IF PROG$(1) = "END" THEN 800 ; IF PROG$(1) = PROG$(1) THEN 140
  350  PRINT "THIS PROGRAM IS USED TO OBTAIN AND SAVE FILES ON PERSONAL DATA FOR"
  360  PRINT "  HUMAN AND RHESUS MONKEY SUBJECTS USED IN THE NBDL EVOKED"
  370  PRINT "   POTENTIAL PROGRAM."
  380  PRINT ; PRINT "BEFORE RUNNING A SUBJECT THE FOLLOWING INFORMATION"
  390  PRINT "  SHOULD BE OBTAINED:"
  400  PRINT ; PRINT "          SUBJECT'S NAME"
  410  PRINT "          SUBJECT'S I.D. NUMBER"
  420  PRINT "          G-FORCE TO BE USED IN EXPERIMENT"
  430  PRINT "          RUN NUMBER FOR EXPERIMENT"
  440  PRINT "          TIME-OF-DAY"
  450  PRINT "          DATE OF EXPERIMENT"
  460  PRINT "          ROOM AIR TEMPERATURE (IN F)"
  470  PRINT "          SUBJECT'S BODY TEMPERATURE (IN F)"
  480  PRINT "          SUBJECT'S HEART RATE (BEATS PER MIN.)"
  490  PRINT "          NUMBER OF CHANNELS TO BE RECORDED"
  500  PRINT "          MONTAGES TO BE USED IN RECORDING"
  510  PRINT; PRINT "BE SURE THAT YOU KNOW THE CORRECT FILE SUBSCRIPT "
  520  PRINT ; PRINT "ANSWER ALL QUESTIONS WITH Y (YES), N (NO), OR A # (N/A),"
```

```
530  PRINT "        UNLESS OTHERWISE REQUESTED.  (NOTE) THE # IS ALSO USED IN"
540  PRINT "        THE 'IS VARIABLE CORRECT' QUESTION."
550  RUN PERDAT
560  PRINT "THIS PROGRAM IS USED TO OBTAIN A DETAILED LISTING OF ALL OF THE"
570  PRINT "     PERSONAL DATA FROM THE PERDAT PROGRAM AND THE PARAMETER"
580  PRINT "     INFORMATION FROM THE EPDATA PROGRAM."
590  RUN LSTPER
600  PRINT "THIS PROGRAM IS USED TO AUTOMATICALLY COLLECT EVOKED POTENTIAL"
610  PRINT "     DATA."
620  PRINT ; PRINT "THE FOLLOWING INFORMATION SHOULD BE KNOWN BEFORE RUNNING"
630  PRINT; PRINT "          THE STIMULUS DELAY TIME"
640  PRINT "          NUMBER OF STIMULI PER SECOND"
650  PRINT "          NUMBER OF SWEEPS TO BE COLLECTED"
660  PRINT "          NUMBER OF CHANNELS TO BE RECORDED"
670  PRINT "          SUBJECT'S I.D. NUMBER"
680  PRINT "          LENGTH OF SWEEP"
690  RUN EPDATA
700  PRINT ; PRINT "THIS PROGRAM IS USED TO TRANSFER PERSONAL, EVOKED"
710  PRINT "     POTENTIAL, OR PARAMETER DATA FROM THE MED-80 TO THE ECLIPSE."
720  RUN MEDECL
730  PRINT "THIS PROGRAM IS FOR THE PLOTTING OF EVOKED POTENTIAL DATA ONTO THE"
740  PRINT "      ZETA PLOTTER."
750  PRINT ; PRINT "WHEN STANDING IN FRONT OF THE ZETA PLOTTER THE PEN'
760  PRINT "     SHOULD BE IN THE UPPER RIGHT HAND CORNER."
770  PRINT ; PRINT "ANSWER ALL QUESTIONS WITH A Y (YES) OR N (NO),"
780  PRINT "        UNLESS OTHERWISE REQUESTED."
790  RUN PLOTEP
800  STOP ; END
$

OK
```

```
LIST PROGBU
$
    10  REM DIMENSION VARIABLES
    20  DIM PROG$(6) , X$(1)
    30  REM ADVANCE PAPER
    40  PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT
    50  PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT
    60  PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT
    70  REM ENABLES USER TO OBTAIN INFORMATION ON A PROGRAM IS THE E.P SERIES
    80  PRINT "DO YOU NEED A LISTING OF THE PROGRAMS BEING OFFERED OR"
    90  INPUT "DO YOU NEED SOME INFORMATION ON THE PURPOSE OF THE PROGRAMS? - ", X$
   100  IF X$(1) = "Y" THEN 120 ; IF X$(1) = "N" THEN 290 ; GOTO 80
   110  REM GENERAL INFORMATION ON A PROGRAM
   120  PRINT ; PRINT "    THE PROGRAM DISK SHOULD BE IN DISK DRIVE # 1."
   130  PRINT "          THE DATA DISK SHOULD BE IN DISK DRIVE # 2."
   140  PRINT ; PRINT " NOTE:  IF YOU MAKE A ERROR NO ERROR MESSAGE IS PRINTED."
   150  PRINT "      THE QUESTION WILL JUST BE REASKED."
   160  PRINT ; PRINT
   170  REM LISTING OF PROGRAMS AND NAMES USED TO CALL THEM
   180  PRINT " THE FOLLOWING IS A LISTING OF ALL PROGRAMS IN THE EVOKED POTENTIAL"
   190  PRINT "   SERIES AND THE SYMBOLS TO BE USED WHEN CALLING A PROGRAM."
   200  PRINT ; PRINT "INPUT OF PERSONAL DATA - PERDAT"
   210  PRINT "COLLECTION OF PARAMETER AND EVOKED POTENTIAL DATA - EPDATA"
   220  PRINT "PLOTTING OF THE EVOKED POTENTIAL DATA ON THE ZETA PLOTTER - PLOTEP"
   230  PRINT "OBTAINING A DETAILED LISTING OF ALL EXPERIMENTAL DATA - LSTPER"
   240  PRINT "TRANSFER OF DATA FILES TO THE ECLIPSE - MEDECL"
   260  PRINT ; PRINT "IF YOU DESIRE TO EXIT THE PROGRAM TYPE 'END' AS A "
   270  PRINT "   RESPONSE TO THE FOLLOWING QUESTION."
   280  REM CALL DESIRED PROGRAM
   290  PRINT ; INPUT "INPUT THE SYMBOL OF THE DESIRED PROGRAM - ", PROG$(1)
   300  PRINT "IS ", PROG$(1), " CORRECT?"
   310  INPUT X$(1)
   320  IF X$(1) = "Y" THEN 330 ; GOTO 290
   330  PRINT
   340  REM CONDITIONAL BRANCH TO PROGRAM SELECTED
   350  IF PROG$(1) = "PERDAT" THEN 440
   360  IF PROG$(1) = "LSTPER" THEN 660 ; IF PROG$(1) = "EPDATA" THEN 710
   370  IF PROG$(1) = "MEDECL" THEN 820 ; IF PROG$(1) = "PLOTEP" THEN 860
   380  IF PROG$(1) = "END" THEN 940 ; IF PROG$(1) = PROG$(1) THEN 190
   430  REM INFORMATION ON THE PERSONAL DATA PROGRAM
   440  PRINT "THIS PROGRAM IS USED TO OBTAIN AND SAVE FILES ON PERSONAL DATA FOR"
   450  PRINT "  HUMAN AND RHESUS MONKEY SUBJECTS USED IN THE NBDL EVOKED"
   460  PRINT "   POTENTIAL PROGRAM."
   470  PRINT ; PRINT "BEFORE RUNNING A SUBJECT THE FOLLOWING INFORMATION"
   480  PRINT "  SHOULD BE OBTAINED:"
   490  PRINT ; PRINT "            SUBJECT'S NAME"
   500  PRINT "          SUBJECT'S I.D. NUMBER"
   510  PRINT "          G-FORCE TO BE USED IN EXPERIMENT"
   520  PRINT "          RUN NUMBER FOR EXPERIMENT"
   530  PRINT "          TIME-OF-DAY"
```

```
540  PRINT "       DATE OF EXPERIMENT"
550  PRINT "       ROOM AIR TEMPERATURE (IN F)"
560  PRINT "       SUBJECT'S BODY TEMPERATURE (IN F)"
570  PRINT "       SUBJECT'S HEART RATE (BEATS PER MIN.)"
580  PRINT "       NUMBER OF CHANNELS TO BE RECORDED"
590  PRINT "       MONTAGES TO BE USED IN RECORDING"
600  PRINT; PRINT "BE SURE THAT YOU KNOW THE CORRECT FILE SUBSCRIPT "
610  PRINT ; PRINT "ANSWER ALL QUESTIONS WITH Y (YES), N (NO), OR A # (N/A),"
620  PRINT "     UNLESS OTHERWISE REQUESTED.  (NOTE) THE # IS ALSO USED IN"
630  PRINT "     THE 'IS VARIABLE CORRECT' QUESTION."
640  RUN PERDAT
650  REM INFORMATION ON THE LISTING OF PERSONAL AND PARAMETER FILE VARIABLES
660  PRINT "THIS PROGRAM IS USED TO OBTAIN A DETAILED LISTING OF ALL OF THE"
670  PRINT "   PERSONAL DATA FROM THE PERDAT PROGRAM AND THE PARAMETER"
680  PRINT "   INFORMATION FROM THE EPDATA PROGRAM."
690  RUN LSTPER
700  REM INFORMATION ON THE EVOKED POTENTIAL DATA PROGRAM
710  PRINT "THIS PROGRAM IS USED TO AUTOMATICALLY COLLECT EVOKED POTENTIAL"
720  PRINT "   DATA."
730  PRINT ; PRINT "THE FOLLOWING INFORMATION SHOULD BE KNOWN BEFORE RUNNING"
740  PRINT; PRINT "       THE STIMULUS DELAY TIME"
750  PRINT "       NUMBER OF STIMULI PER SECOND"
760  PRINT "       NUMBER OF SWEEPS TO BE COLLECTED"
770  PRINT "       NUMBER OF CHANNELS TO BE RECORDED"
780  PRINT "       SUBJECT'S I.D. NUMBER"
790  PRINT "       LENGTH OF SWEEP"
800  RUN EPDATA
810  REM INFORMATION ON THE TRANSFER OF DATA PROGRAM
820  PRINT ; PRINT "THIS PROGRAM IS USED TO TRANSFER PERSONAL, EVOKED"
830  PRINT "   POTENTIAL, OR PARAMETER DATA FROM THE MED-80 TO THE ECLIPSE."
840  RUN MEDECL
850  REM INFORMATION ON THE PLOTTING OF EVOKED POTENTIALS PROGRAM
860  PRINT "THIS PROGRAM IS FOR THE PLOTTING OF EVOKED POTENTIAL DATA ONTO THE"
870  PRINT "    ZETA PLOTTER."
880  PRINT ; PRINT "WHEN STANDING IN FRONT OF THE ZETA PLOTTER THE PEN"
890  PRINT "   SHOULD BE IN THE UPPER RIGHT HAND CORNER."
900  PRINT ; PRINT "ANSWER ALL QUESTIONS WITH A Y (YES) OR N (NO),"
910  PRINT "     UNLESS OTHERWISE REQUESTED."
920  RUN PLOTEP
930  REM EXIT PROGRAM
940  STOP ; END
$

OK
```

```
LIST PERDAT
$
   10  DIM RCH$(1),BLANK$(60),XX$(60),X$(1),CH$(2),NAME$(35),COMNT$(60)
   20  DIM SUBNO$(5),DATE$(9),GFORC$(6),RUNNO$(7),TIMDA$(6),AIRTP$(6),SUBTP$(5)
   30  DIM HEART$(4),INA$(5),INA1$(5),INA2$(5),INA3$(5),INA4$(5),INA5$(5),INA6$(5)
   40  DIM INA7$(5),INA8$(5),INB$(5),INB1$(5),INB2$(5),INB3$(5),INB4$(5),INB5$(5)
   50  DIM INB6$(5) , INB7$(5) , INB8$(5) , Y$(230)
   60  N1(1) = 0 ; N2(1) = 0 ; N3(1) = 0 ; N4(1) = 0 ; W = 1 ; Y$ = " "
   70  BLANK$ = "                                                            "
   80  PRINT ; PRINT "INPUT NAME AS FOLLOWS."
   90  P = 35 ; GOSUB 1930
  100  INPUT "LAST, FIRST, THEN M.I. - ", NAME$(1)
  110  XX$(1) = NAME$(1) ; GOSUB 2060
  120  IF X$(1) = "Y" THEN 130 ; IF X$ = "#" THEN 140 ; GOTO 100
  130  Y$(W) = BLANK$(1) ; Y$(W) = NAME$(1)
  140  W = W + P ; P = 5 ; GOSUB 1930
  150  PRINT ; PRINT "H = HUMAN SUBJECT       R = RHESUS SUBJECT"
  160  INPUT "SUBJECT'S I.D. NUMBER - ", SUBNO$(1)
  170  XX$(1) = SUBNO$(1) ; GOSUB 2060
  180  IF X$(1) = "Y" THEN 190 ; IF X$(1) = "#" THEN 200 ; GOTO 150
  190  Y$(W) = BLANK$(1) ; Y$(W) = SUBNO$(1)
  200  W = W + P ; P = 9 ; GOSUB 1930
  210  INPUT "MON/DAY/YEAR - ", DATE$(1)
  220  XX$(1) = DATE$(1) ; GOSUB 2060
  230  IF X$(1) = "Y" THEN 240 ; IF X$(1) = "#" THEN 250 ; GOTO 210
  240  Y$(W) = BLANK$(1) ; Y$(W) = DATE$(1)
  250  W = W + P ; P = 6 ; GOSUB 1930
  260  PRINT;PRINT "IF G-FORCE IS BELOW 10 ENTER THE # WITH A 0 FIRST (I.E. 05G+X)
  270  INPUT "G-FORCE - ", GFORC$(1)
  280  XX$(1) = GFORC$(1) ; GOSUB 2060
  290  IF X$(1) = "Y" THEN 300 ; IF X$(1) = "#" THEN 340 ; GOTO 260
  300  IF GFORC$(4,4) = "+" OR GFORC$(4,4) = "-" THEN 310 ; GOTO 260
  310  IF GFORC$(5,5) = "X" OR GFORC$(5,5) = "Y" OR GFORC$(5,5) = "Z" THEN 330
  320  IF GFORC$(1) = GFORC$(1) THEN 260
  330  Y$(W) = BLANK$(1) ; Y$(W) = GFORC$(1)
  340  W = W + P ; P = 7 ; GOSUB 1930
  350  INPUT "RUN NUMBER - ", RUNNO$(1)
  360  XX$(1) = RUNNO$(1) ; GOSUB 2060
  370  IF X$(1) = "Y" THEN 380 ; IF X$(1) = "#" THEN 390 ; GOTO 350
  380  Y$(W) = BLANK$(1) ; Y$(W) = RUNNO$(1)
  390  W = W + P ; P = 6 ; GOSUB 1930
  400  INPUT "TIME-OF-DAY ON A 24 HOUR CLOCK (I.E. 14:33) - ", TIMDA$(1)
  410  XX$(1) = TIMDA$(1) ; GOSUB 2060
  420  IF X$(1) = "Y" THEN 430 ; IF X$(1) = "#" THEN 440 ; GOTO 400
  430  Y$(W) = BLANK$(1) ; Y$(W) = TIMDA$(1)
  440  W = W + P ; P = 5 ; GOSUB 1930
  450  INPUT "AIR TEMPERATURE (IN F) - ", AIRTP$(1)
  460  XX$(1) = AIRTP$(1) ; GOSUB 2060
  470  IF X$(1) = "Y" THEN 480 ; IF X$(1) = "#" THEN 500 ; GOTO 450
  480  IF AIRTP$(1) < "69" OR AIRTP$(1) > "81" THEN 450
```

```
490  Y$(W) = BLANK$(1) ; Y$(W) = AIRTP$(1)
500  W = W + P ; P = 5 ; GOSUB 1930
510  INPUT "SUBJECT'S TEMPERATURE (IN F) - ", SUBTP$(1)
520  XX$(1) = SUBTP$(1) ; GOSUB 2060
530  IF X$(1) = "Y" THEN 540 ; IF X$(1) = "#" THEN 550 ; GOTO 510
540  Y$(W) = BLANK$(1) ; Y$(W) = SUBTP$(1)
550  W = W + P ; P = 4 ; GOSUB 1930
560  INPUT "SUBJECT'S HEART RATE (BEATS/MIN.) - ", HEART$(1)
570  XX$(1) = HEART$(1) ; GOSUB 2060
580  IF X$(1) = "Y" THEN 590 ; IF X$(1) = "#" THEN 600 ; GOTO 560
590  Y$(W) = BLANK$(1) ; Y$(W) = HEART$(1)
600  W = W + P ; P = 60 ; GOSUB 1930
610  INPUT "PRINT COMMENTS (USE LESS THAN 60 CHARACTERS) - ", COMNT$(1)
620  XX$(1) = COMNT$(1) ; GOSUB 2060
630  IF X$(1) = "Y" THEN 640 ; IF X$(1) = "#" THEN 650 ; GOTO 610
640  Y$(W) = BLANK$(1) ; Y$(W) = COMNT$(1)
650  W = W + P ; P = 2 ; GOSUB 1930
660  INPUT "NUMBER OF CHANNELS BEING RECORDED (1 - 8) - ", CH$(1)
670  XX$(1) = CH$(1) ; GOSUB 2060
680  IF X$(1) = "Y" THEN 690 ; IF X$(1) = "#" THEN 710 ; GOTO 660
690  IF CH$(1) < "1" OR CH$(1) > "8" THEN 660
700  Y$(W) = BLANK$(1) ; Y$(W) = CH$(1)
710  W = W + P
720  IF CH$(1) = "1" THEN 1380 ; IF CH$(1) = "2" THEN 1290
730  IF CH$(1) = "3" THEN 1200 ; IF CH$(1) = "4" THEN 1110
740  IF CH$(1) = "5" THEN 1020 ; IF CH$(1) = "6" THEN 930
750  IF CH$(1) = "7" THEN 840
760  P = 5 ; GOSUB 1930
770  PRINT "INPUT MONTAGE FOR CHANNEL 8"
780  INPUT "   INPUT A - ", INA8$(1)
790  Y$(W) = BLANK$(1) ; Y$(W) = INA8$(1)
800  W = W + P ; P = 5 ; GOSUB 1930
810  INPUT "   INPUT B - ", INB8$(1)
820  Y$(W) = BLANK$(1) ; Y$(W) = INB8$(1)
830  W = W + P ; GOTO 850
840  P = 10 ; GOSUB 1930 ; W = W + P
850  PRINT "INPUT MONTAGE FOR CHANNEL 7"
860  P = 5 ; GOSUB 1930
870  INPUT "   INPUT A - ", INA7$(1)
880  Y$(W) = BLANK$(1) ; Y$(W) = INA7$(1)
890  W = W + P ; P = 5 ; GOSUB 1930
900  INPUT "   INPUT B - ", INB7$(1)
910  Y$(W) = BLANK$(1) ; Y$(W) = INB7$(1)
920  W = W + P ; GOTO 940
930  P = 20 ; GOSUB 1930 ; W = W + P
940  PRINT "INPUT MONTAGE FOR CHANNEL 6"
950  P = 5 ; GOSUB 1930
960  INPUT "   INPUT A - ", INA6$(1)
970  Y$(W) = BLANK$(1) ; Y$(W) = INA6$(1)
980  W = W + P ; P = 5 ; GOSUB 1930
990  INPUT "   INPUT B - ", INB6$(1)
```

```
1000  Y$(W) = BLANK$(1) ; Y$(W) = INB6$(1)
1010  W = W + P ; GOTO 1030
1020  P = 30 ; GOSUB 1930 ; W = W + P
1030  PRINT "INPUT MONTAGE FOR CHANNEL 5"
1040   P = 5 ; GOSUB 1930
1050  INPUT "   INPUT A - ", INA5$(1)
1060  Y$(W) = BLANK$(1) ; Y$(W) = INA5$(1)
1070  W = W + P ; P = 5 ; GOSUB 1930
1080  INPUT "   INPUT B - ", INB5$(1)
1090  Y$(W) = BLANK$(1) ; Y$(W) = INB5$(1)
1100  W = W + P ; GOTO 1120
1110  P = 40 ; GOSUB 1930 ; W = W + P
1120  PRINT "INPUT MONTAGE FOR CHANNEL 4"
1130  P = 5 ; GOSUB 1930
1140  INPUT "   INPUT A - ", INA4$(1)
1150  Y$(W) = BLANK$(1) ; Y$(W) = INA4$(1)
1160  W = W + P ; P = 5 ; GOSUB 1930
1170  INPUT "   INPUT B - ", INB4$(1)
1180  Y$(W) = BLANK$(1) ; Y$(W) = INB4$(1)
1190  W = W + P ; GOTO 1210
1200  P = 50 ; GOSUB 1930 ; W = W + P
1210  PRINT "INPUT MONTAGE FOR CHANNEL 3"
1220  P = 5 ; GOSUB 1930
1230  INPUT "   INPUT A - ", INA3$(1)
1240  Y$(W) = BLANK$(1) ; Y$(W) = INA3$(1)
1250  W = W + P ; P = 5 ; GOSUB 1930
1260  INPUT "   INPUT B - ", INB3$(1)
1270  Y$(W) = BLANK$(1) ; Y$(W) = INB3$(1)
1280  W = W + P ; GOTO 1300
1290  P = 60 ; GOSUB 1930
1300  PRINT "INPUT MONTAGE FOR CHANNEL 2"
1310  P = 5 ; GOSUB 1930
1320  INPUT "   INPUT A - ", INA2$(1)
1330  Y$(W) = BLANK$(1) ; Y$(W) = INA2$(1)
1340  W = W + P ; P = 5 ; GOSUB 1930
1350  INPUT "   INPUT B - ", INB2$(1)
1360  Y$(W) = BLANK$(1) ; Y$(W) = INB2$(1)
1370  W = W + P ; GOTO 1390
1380  P = 70 ; GOSUB 1930 ; W = W + P
1390  PRINT "INPUT MONTAGE FOR CHANNEL 1"
1400  P = 5 ; GOSUB 1930
1410  INPUT "   INPUT A - ", INA1$(1)
1420  Y$(W) = BLANK$(1) ; Y$(W) = INA1$(1)
1430  W = W + P ; P = 5 ; GOSUB 1930
1440  INPUT "   INPUT B - ", INB1$(1)
1450  Y$(W) = BLANK$(1) ; Y$(W) = INB1$(1)
1460  W = W + P ; P = 5 ; GOSUB 1930
1470  PRINT ; PRINT "  CH. #", "INPUT A", "INPUT B"
1480  SS(1) = 225
1490  FOR S = 1 TO 8
1500  SS(1) = SS(1) - 10
```

```
1510  PRINT S IN 4,4 , Y$(SS(1),SS(1)+4), Y$(SS(1)+5,SS(1)+9)
1520  NEXT S
1530  PRINT ; INPUT "ARE THERE ANY ERRORS IN THE ABOVE - ", X$
1540  IF X$ = "Y" THEN 1550 ; IF X$ = "N" THEN 1720 ; GOTO 1530
1550  INPUT "WHICH CHANNEL IS INCORRECT - ", RCH$(1)
1560  PRINT "IS ", RCH$(1), " CORRECT?"
1570  INPUT X$(1)
1580  IF X$(1) = "Y" THEN 1590 ; GOTO 1550
1590  IF RCH$(1) = "1" THEN 1640 ; IF RCH$(1) = "2" THEN 1650
1600  IF RCH$(1) = "3" THEN 1660 ; IF RCH$(1) = "4" THEN 1670
1610  IF RCH$(1) = "5" THEN 1680 ; IF RCH$(1) = "6" THEN 1690
1620  IF RCH$(1) = "7" THEN 1700 ; IF RCH$(1) = "8" THEN 1710
1630  IF RCH$(1) = RCH$(1) THEN 1550
1640  N1(1)=215 ; N2(1)=219 ; N3(1)=220 ; N4(1)=224 ; GOSUB 1970 ; GOTO 1470
1650  N (1)=205 ; N2(1)=209 ; N3(1)=210 ; N4(1)=214 ; GOSUB 1970 ; GOTO 1470
1660  N1(1)=195 ; N2(1)=199 ; N3(1)=200 ; N4(1)=204 ; GOSUB 1970 ; GOTO 1470
1670  N1(1)=185 ; N2(1)=189 ; N3(1)=190 ; N4(1)=194 ; GOSUB 1970 ; GOTO 1470
1680  N1(1)=175 ; N2(1)=179 ; N3(1)=180 ; N4(1)=184 ; GOSUB 1970 ; GOTO 1470
1690  N1(1)=165 ; N2(1)=169 ; N3(1)=170 ; N4(1)=174 ; GOSUB 1970 ; GOTO 1470
1700  N1(1)=155 ; N2(1)=159 ; N3(1)=160 ; N4(1)=164 ; GOSUB 1970 ; GOTO 1470
1710  N1(1)=145 ; N2(1)=149 ; N3(1)=150 ; N4(1)= 54 ; GOSUB 1970 ; GOTO 1470
1720  PRINT ; INPUT "STORE THIS PERSONAL DATA - ", X$
1730  IF X$ = "Y" THEN 1740 ; IF X$ = "N" THEN 1890 ; GOTO 1720
1740  PUT 2 IN PDV ; PUT 0 IN PST ; PUT 1024 IN PSZ
1750  PRINT ; INPUT "INPUT FILE SUBSCRIPT (3 #'S 0 - 999) - ", SUB
1760  PRINT "IS ", SUB , " CORRECT?"
1770  INPUT X$
1780  IF X$ = "Y" THEN 1790 ; GOTO 1750
1790  IF SUB < 0 OR SUB > 999 THEN 1750 ; PUT SUB IN PFS
1800  BASE = VAL(SUBNO$(3,4)) ; PUT BASE IN PFB
1810  HEAD = ASC(SUBNO$(1)) ; IF HEAD = 48 THEN 1820 ; PUT 210 IN PFH ; GOTO 1830
1820  PUT 200 IN PFH
1830  FOR S = 1 TO W
1840  A$ = Y$(S,S) ; DMEM(S) = ASC(A$) ; NEXT S
1850  XEQ DWR
1860  PRINT ; INPUT "COLLECT EVOKED POTENTIAL DATA NOW - ", X$
1870  IF X$ = "Y" THEN 1880 ; IF X$ = "N" THEN 1890 ; GOTO 1860
1880  RUN EPDATA
1890  PRINT ; INPUT "RUN ANOTHER PROGRAM - ", X$
1900  IF X$ = "Y" THEN 1910 ; IF X$ = "N" THEN 2090 ; GOTO 1890
1910  RUN EPPROG
1920  STOP
1930  FOR S = W TO W + (P - 1)
1940  Y$(S,S) = "#"
1950  NEXT S
1960  RETURN ; STOP
1970  INPUT "CORRECT 'INPUT A' VALUE - ", INA$(1)
1980  XX$(1) = INA$(1) ; GOSUB 2060
1990  IF X$(1) = "Y" THEN 2000 ; GOTO 1970
2000  Y$(N1(1),N2(1)) = INA$(1)
2010  INPUT "CORRECT 'INPUT B' VALUE - ", INB$(1)
```

```
 2020  XX$(1) = INB$(1) ; GOSUB 2060
 2030  IF X$(1) = "Y" THEN 2040 ; GOTO 2010
 2040  Y$(N3(1),N4(1)) = INB$(1)
 2050  RETURN ; STOP
 2060  PRINT "IS ", XX$(1), " CORRECT?"
 2070  INPUT X$(1)
 2080  RETURN ; STOP
 2090  END
$
```

```
OK
```

```
LIST PERBU
$
     5  REM DIMENSION VARIABLES.
    10  DIM RCH$(1),Y$(225),BLANK$(60),XX$(60),X$(1),CH$(2),NAME$(35),COMNT$(60)
    20  DIM SUBNO$(5),DATE$(9),GFORC$(6),RUNNO$(7),TIMDA$(6),AIRTP$(6),SUBTP$(5)
    30  DIM HEART$(4),FROM$(5),FROM1$(5),FROM2$(5),FROM3$(5),FROM4$(5),FROM5$(5)
    40  DIM FROM6$(5),FROM7$(5),FROM8$(5),TO$(5),TO1$(5),TO2$(5),TO3$(5),TO4$(5)
    50  DIM TO5$(5),TO6$(5),TO7$(5),TO8$(5)
    55  REM ASSIGN VALUES TO SOME VARIABLES.
    60  N1(1) = 0 ; N2(1) = 0 ; N3(1) = 0 ; N4(1) = 0 ; W = 1 ; Y$ = " "
    70  BLANK$ = "                                                            "
    75  REM FOR THE FOLLOWING VARIABLES USER INPUTS THEIR VALUES FOR THIS EXPERIMENT
    76  REM  A VISUAL CHECK ON THE VALUE IS PROVIDED.  AFTER THE CORRECT VALUE HAS
    77  REM  BEEN ENTERED THE VARIABLE IS STORED IN THE PERSONAL DATA FILE IN ITS
    78  REM  CORRECT LOCATION, REPRESENTED BY 'W'.  THIS LOCATION IS INCREASED BY THE
    79  REM  SIZE OF THE VAR UNDER CONSIDERATION 'P', AND PROG ADVANCES TO NEXT VAR.
    80  PRINT ; PRINT "INPUT NAME AS FOLLOWS."
    90  P = 35 ; GOSUB 1930
   100  INPUT "LAST, FIRST, THEN M.I. - ", NAME$(1)
   110  XX$(1) = NAME$(1) ; GOSUB 2060
   120  IF X$(1) = "Y" THEN 130 ; IF X$ = "#" THEN 140 ; GOTO 100
   130  Y$(W) = BLANK$(1) ; Y$(W) = NAME$(1)
   140  W = W + P ; P = 5 ; GOSUB 1930
   150  PRINT ; PRINT "H = HUMAN SUBJECT      R = RHESUS SUBJECT"
   160  INPUT "SUBJECT'S I.D. NUMBER - ", SUBNO$(1)
   170  XX$(1) = SUBNO$(1) ; GOSUB 2060
   180  IF X$(1) = "Y" THEN 190 ; IF X$(1) = "#" THEN 200 ; GOTO 150
   190  Y$(W) = BLANK$(1) ; Y$(W) = SUBNO$(1)
   200  W = W + P ; P = 9 ; GOSUB 1930
   210  INPUT "MON/DAY/YEAR - ", DATE$(1)
   220  XX$(1) = DATE$(1) ; GOSUB 2060
   230  IF X$(1) = "Y" THEN 240 ; IF X$(1) = "#" THEN 250 ; GOTO 210
   240  Y$(W) = BLANK$(1) ; Y$(W) = DATE$(1)
   250  W = W + P ; P = 6 ; GOSUB 1930
   260  PRINT;PRINT "IF G-FORCE IS BELOW 10 ENTER THE # WITH A 0 FIRST (I.E. 03G-Y)"
   270  INPUT "G-FORCE - ", GFORC$(1)
   280  XX$(1) = GFORC$(1) ; GOSUB 2060
   290  IF X$(1) = "Y" THEN 300 ; IF X$(1) = "#" THEN 340 ; GOTO 260
   300  IF GFORCE$(4,4) = "+" OR GFORC$(4,4) = "-" THEN 310 ; GOTO 260
   310  IF GFORC$(5,5) = "X" OR GFORC$(5,5) = "Y" OR GFORC$(5,5) = "Z" THEN 330
   320  IF GFORC$(1) = GFORC$(1) THEN 260
   330  Y$(W) = BLANK$(1) ; Y$(W) = GFORC$(1)
   340  W = W + P ; P = 7 ; GOSUB 1930
   350  INPUT "RUN NUMBER - ", RUNNO$(1)
   360  XX$(1) = RUNNO$(1) ; GOSUB 2060
   370  IF X$(1) = "Y" THEN 380 ; IF X$(1) = "#" THEN 390 ; GOTO 350
   380  Y$(W) = BLANK$(1) ; Y$(W) = RUNNO$(1)
   390  W = W + P ; P = 6 ; GOSUB 1930
   400  INPUT "TIME-OF-DAY ON A 24 HOUR CLOCK (I.E. 14:33) - ", TIMDA$(1)
   410  XX$(1) = TIMDA$(1) ; GOSUB 2060
```

```
420  IF X$(1) = "Y" THEN 430 ; IF X$(1) = "#" THEN 440 ; GOTO 400
430  Y$(W) = BLANK$(1) ; Y$(W) = TIMDA$(1)
440  W = W + P ; P = 5 ; GOSUB 1930
450  INPUT "AIR TEMPERATURE (IN F) - ", AIRTP$(1)
460  XX$(1) = AIRTP$(1) ; GOSUB 2060
470  IF X$(1) = "Y" THEN 480 ; IF X$(1) = "#" THEN 500 ; GOTO 450
475  REM CHECK VALUE OF AIR TEMPERATURE.
480  IF AIRTP$(1) < "69" OR AIRTP$(1) > "81" THEN 450
490  Y$(W) = BLANK$(1) ; Y$(W) = AIRTP$(1)
500  W = W + P ; P = 5 ; GOSUB 1930
510  INPUT "SUBJECT'S TEMPERATURE (IN F) - ", SUBTP$(1)
520  XX$(1) = SUBTP$(1) ; GOSUB 2060
530  IF X$(1) = "Y" THEN 540 ; IF X$(1) = "#" THEN 550 ; GOTO 510
540  Y$(W) = BLANK$(1) ; Y$(W) = SUBTP$(1)
550  W = W + P ; P = 4 ; GOSUB 1930
560  INPUT "SUBJECT'S HEART RATE (BEATS/MIN.) - ", HEART$(1)
570  XX$(1) = HEART$(1) ; GOSUB 2060
580  IF X$(1) = "Y" THEN 590 ; IF X$(1) = "#" THEN 600 ; GOTO 560
590  Y$(W) = BLANK$(1) ; Y$(W) = HEART$(1)
600  W = W + P ; P = 60 ; GOSUB 1930
610  INPUT "PRINT COMMENTS (USE LESS THAN 60 CHARACTERS) - ", COMNT$(1)
620  XX$(1) = COMNT$(1) ; GOSUB 2060
630  IF X$(1) = "Y" THEN 640 ; IF X$(1) = "#" THEN 650 ; GOTO 610
640  Y$(W) = BLANK$(1) ; Y$(W) = COMNT$(1)
650  W = W + P ; P = 2 ; GOSUB 1930
660  INPUT "NUMBER OF CHANNELS BEING RECORDED (1 - 8) - ", CH$(1)
670  XX$(1) = CH$(1) ; GOSUB 2060
680  IF X$(1) = "Y" THEN 690 ; IF X$(1) = "#" THEN 710 ; GOTO 660
685  REM CHECK VALUE OF THE NUMBER OF CHANNELS.
690  IF CH$(1) < "1" OR CH$(1) > "8" THEN 660
700  Y$(W) = BLANK$(1) ; Y$(W) = CH$(1)
710  W = W + P
715  REM INPUT A CHANNELS MONTAGE FOR THE EXPERIMENT.
720  IF CH$(1) = "1" THEN 1380 ; IF CH$(1) = "2" THEN 1290
730  IF CH$(1) = "3" THEN 1200 ; IF CH$(1) = "4" THEN 1110
740  IF CH$(1) = "5" THEN 1020 ; IF CH$(1) = "6" THEN 930
750  IF CH$(1) = "7" THEN 840
760  P = 5 ; GOSUB 1930
770  PRINT "INPUT MONTAGE FOR CHANNEL 8"
780  INPUT "  INPUT A - ", FROM8$(1)
790  Y$(W) = BLANK$(1) ; Y$(W) = FROM8$(1)
800  W = W + P ; P = 5 ; GOSUB 1930
810  INPUT "   INPUT B - ", TO8$(1)
820  Y$(W) = BLANK$(1) ; Y$(W) = TO8$(1)
830  W = W + P ; GOTO 850
840  P = 10 ; GOSUB 1930 ; W = W + P
850  PRINT "INPUT MONTAGE FOR CHANNEL 7"
860  P = 5 ; GOSUB 1930
870  INPUT "   INPUT A - ", FROM7$(1)
880  Y$(W) = BLANK$(1) ; Y$(W) = FROM7$(1)
890  W = W + P ; P = 5 ; GOSUB 1930
```

```
 900  INPUT "   INPUT B - ", TO7$(1)
 910  Y$(W) = BLANK$(1) ; Y$(W) = TO7$(1)
 920  W = W + P ; GOTO 940
 930  P = 20 ; GOSUB 1930 ; W = W + P
 940  PRINT "INPUT MONTAGE FOR CHANNEL 6"
 950  P = 5 ; GOSUB 1930
 960  INPUT "   INPUT A - ", FROM6$(1)
 970  Y$(W) = BLANK$(1) ; Y$(W) = FROM6$(1)
 980  W = W + P ; P = 5 ; GOSUB 1930
 990  INPUT "   INPUT B - ", TO6$(1)
1000  Y$(W) = BLANK$(1) ; Y$(W) = TO6$(1)
1010  W = W + P ; GOTO 1030
1020  P = 30 ; GOSUB 1930 ; W = W + P
1030  PRINT "INPUT MONTAGE FOR CHANNEL 5"
1040  P = 5 ; GOSUB 1930
1050  INPUT "   INPUT A - ", FROM5$(1)
1060  Y$(W) = BLANK$(1) ; Y$(W) = FROM5$(1)
1070  W = W + P ; P = 5 ; GOSUB 1930
1080  INPUT "   INPUT B - ", TO5$(1)
1090  Y$(W) = BLANK$(1) ; Y$(W) = TO5$(1)
1100  W = W + P ; GOTO 1120
1110  P = 40 ; GOSUB 1930 ; W = W + P
1120  PRINT "INPUT MONTAGE FOR CHANNEL 4"
1130  P = 5 ; GOSUB 1930
1140  INPUT "   INPUT A - ", FORM4$(1)
1150  Y$(W) = BLANK$(1) ; Y$(W) = FROM4$(1)
1160  W = W + P ; P = 5 ; GOSUB 1930
1170  INPUT "   INPUT B - ", TO4$(1)
1180  Y$(W) = BLANK$(1) ; Y$(W) = TO4$(1)
1190  W = W + P ; GOTO 1210
1200  P = 50 ; GOSUB 1930 ; W = W + P
1210  PRINT "INPUT MONTAGE FOR CHANNEL 3"
1220  P = 5 ; GOSUB 1930
1230  INPUT "   INPUT A - ", FROM3$(1)
1240  Y$(W) = BLANK$(1) ; Y$(W) = FROM3$(1)
1250  W = W + P ; P = 5 ; GOSUB 1930
1260  INPUT "   INPUT B - ", TO3$(1)
1270  Y$(W) = BLANK$(1) ; Y$(W) = TO3$(1)
1280  W = W + P ; GOTO 1300
1290  P = 60 ; GOSUB 1930
1300  PRINT "INPUT MONTAGE FOR CHANNEL 2"
1310  P = 5 ; GOSUB 1930
1320  INPUT "   INPUT A - ", FROM2$(1)
1330  Y$(W) = BLANK$(1) ; Y$(W) = FROM2$(1)
1340  W = W + P ; P = 5 ; GOSUB 1930
1350  INPUT "   INPUT B - ", TO2$(1)
1360  Y$(W) = BLANK$(1) ; Y$(W) = TO2$(1)
1370  W = W + P ; GOTO 1390
1380  P = 70 ; GOSUB 1930 ; W = W + P
1390  PRINT "INPUT MONTAGE FOR CHANNEL 1"
1400  P = 5 ; GOSUB 1930
```

```
1410  INPUT "   INPUT A - ", FORM1$(1)
1420  Y$(W) = BLANK$(1) ; Y$(W) = FROM1$(1)
1430  W = W + P ; P = 5 ; GOSUB 1930
1440  INPUT "   INPUT B - ", TO1$(1)
1450  Y$(W) = BLANK$(1) ; Y$(W) = TO1$(1)
1460  W = W + P ; P = 5 ; GOSUB 1930
1465  REM PRINT OUT THE MONTAGE INFORMATION FOR ALL CHANNELS.
1470  PRINT ; PRINT "  CH. #", "INPUT A", "INPUT B"
1480  SS(1) = 225
1490  FOR I = 1 TO 8
1500  SS(1) = SS(1) - 10
1510  PRINT I IN 4,4 , Y$(SS(1),SS(1)+4), Y$(SS(1)+5,SS(1)+9)
1520  NEXT I
1525  REM ENABLES USER TO CHANGE A MONTAGES VALUE.
1530  PRINT ; INPUT "ARE THE ABOVE MONTAGES CORRECT - ", X$
1540  IF X$ = "Y" THEN 1720 ; IF X$ = "N" THEN 1550 ; GOTO 1530
1545  REM USER SELECTS CHANNEL WITH ERRORS IN ITS MONTAGE VALUES.
1550  INPUT "WHICH CHANNEL IS INCORRECT - ", RCH$(1)
1560  PRINT "IS ", RCH$(1), " CORRECT?"
1570  INPUT X$(1)
1580  IF X$(1) = "Y" THEN 1590 ; GOTO 1550
1585  REM REPLACEMENT OF THE BAD MONTAGES WITH THE CORRECT VALUE.
1590  IF RCH$(1) = "1" THEN 1640 ; IF RCH$(1) = "2" THEN 1650
1600  IF RCH$(1) = "3" THEN 1660 ; IF RCH$(1) = "4" THEN 1670
1610  IF RCH$(1) = "5" THEN 1680 ; IF RCH$(1) = "6" THEN 1690
1620  IF RCH$(1) = "7" THEN 1700 ; IF RCH$(1) = "8" THEN 1710
1630  IF RCH$(1) = RCH$(1) THEN 1550
1640  N1(1)=215 ; N2(1)=219 ; N3(1)=220 ; N4(1)=224 ; GOSUB 1970 ; GOTO 1470
1650  N1(1)=205 ; N2(1)=209 ; N3(1)=210 ; N4(1)=214 ; GOSUB 1970 ; GOTO 1470
1660  N1(1)=195 ; N2(1)=199 ; N3(1)=200 ; N4(1)=204 ; GOSUB 1970 ; GOTO 1470
1670  N1(1)=185 ; N2(1)=189 ; N3(1)=190 ; N4(1)=194 ; GOSUB 1970 ; GOTO 1470
1680  N1(1)=175 ; N2(1)=179 ; N3(1)=180 ; N4(1)=184 ; GOSUB 1970 ; GOTO 1470
1690  N1(1)=165 ; N2(1)=169 ; N3(1)=170 ; N4(1)=174 ; GOSUB 1970 ; GOTO 1470
1700  N1(1)=155 ; N2(1)=159 ; N3(1)=160 ; N4(1)=164 ; GOSUB 1970 ; GOTO 1470
1710  N1(1)=145 ; N2(1)=149 ; N3(1)=150 ; N4(1)=154 ; GOSUB 1970 ; GOTO 1470
1715  REM ENABLES USER TO STORE THE PERSONAL DATA FILE
1720  PRINT ; INPUT "STORE THIS PERSONAL DATA - ", X$
1730  IF X$ = "Y" THEN 1740 ; IF X$ = "N" THEN 1890 ; GOTO 1720
1735  REM SELECTS DISK DRIVE NUMBER 2 AND SETS THE START AND SIZE OF DMEM.
1740  PUT 2 IN PDV ; PUT 0 IN PST ; PUT 1024 IN PSZ
1750  PRINT ; INPUT "INPUT FILE SUBSCRIPT (3 #'S 0 - 999) - ", S
1760  PRINT "IS ", S, " CORRECT?"
1770  INPUT X$
1780  IF X$ = "Y" THEN 1790 ; GOTO 1750
1785  REM CHECK TO INSURE THAT THE FILE SUBSCRIPT IS IN THE CORRECT RANGE.
1790  IF S < 0 OR S > 999 THEN 1750 ; PUT S IN PFS
1795  REM SELECTION OF THE FILE BASE
1800  BASE = VAL(SUBNO$(3,4)) ; PUT BASE IN PFB
1805  REM SECLECTION OF THE FILE HEADER.
1810  HEAD = ASC(SUBNO$(1)) ; IF HEAD = 40 THEN 1820 ; PUT 210 IN PFH ; GOTO 1830
1820  PUT 200 IN PFH
```

```
1825  REM CHANGE NORMAL DATA INTO ASCII CODE.
1830  FOR I = 1 TO W
1840  A$ = Y$(I,I) ; DMEM(I) = ASC(A$) ; NEXT I
1845  REM WRITE FILE ON DISK.
1850  XEQ DWR
1855  REM ENABLES USER TO GO DIRECTLY TO THE PROGRAM WHICH COLLECTS E.P. DATA
1860  PRINT ; INPUT "COLLECT EVOKED POTENTIAL DATA NOW - ", X$
1870  IF X$ = "Y" THEN 1880 ; IF X$ = "N" THEN 1890 ; GOTO 1860
1880  RUN EPDATA
1885  REM ENABLES USER TO RETURN TO THE POINTER PROG. AND RUN ANOTHER PROGRAM
1890  PRINT ; INPUT "RUN ANOTHER PROGRAM - ", X$
1900  IF X$ = "Y" THEN 1910 ; IF X$ = "N" THEN 2090 ; GOTO 1890
1910  RUN EPPROG
1915  REM STOP THE FOLLOWING ARE SUBROUTINES.
1920  STOP
1925  REM PLACES A NULL FACTOR '#' IN A VARIABLE'S LOCATION.
1930  FOR S = W TO W + (P - 1)
1940  Y$(S,S) = "#"
1950  NEXT S
1960  RETURN ; STOP
1965  REM ENABLES USER TO CHANGE A CHANNEL'S MONTAGE VALUE.
1970  INPUT "CORRECT 'INPUT A' MONTAGE - ", FROM$(1)
1980  XX$(1) = FROM$(1) ; GOSUB 2060
1990  IF X$(1) = "Y" THEN 2000 ; GOTO 1970
2000  Y$(N1(1),N2(1)) = FROM$(1)
2010  INPUT "CORRECT 'INPUT B' MONTAGE - ", TO$(1)
2020  XX$(1) = TO$(1) ; GOSUB 2060
2030  IF X$(1) = "Y" THEN 2040 ; GOTO 2010
2040  Y$(N3(1),N4(1)) = TO$(1)
2050  RETURN ; STOP
2055  REM ENABLES USER TO RESPOND TO A VISUAL CHECK ON A VARIABLES VALUE.
2060  PRINT "IS ", XX$(1), " CORRECT?"
2070  INPUT X$(1)
2080  RETURN ; STOP
2085  REM END OF THE PROGRAM
2090  END
$

OK
```

```
LIST EPDATA
$
    10  PUT 0 IN PST ; PUT 8192 IN PSZ ; XEQ ZEM
    20  DIM DELAY$(3), STSEC$(3), SWPSZ$(7), SWPS$(5), CHS$(2), STMEM$(2), WRIST$(6)
    30  DIM SZMEM$(5) , DWELL$(5), STPER$(6) , REPTS$(5), Y$(70) , A$(70)
    40  DIM LBAND$(4), HBAND$(5), SENTY$(5), STINT$(5), THRES$(4), SUBNO$(5), VAL$(7)
    50  DIM LLEWD$(5) , X$(1) ; N1(1) = 0 ; N2(1) = 0
    60  W = 1 ; SWPSZ(1) = 0 ; GOTO 70
    70  SUBNO$(1) = "     "
    80  PRINT ; INPUT "INPUT THE SUBJECT'S I.D. NUMBER - ", SUBNO$(1)
    90  PRINT "IS ", SUBNO$(1), " CORRECT?"
   100  INPUT X$
   110  IF X$ = "Y" THEN 120 ; IF X$ = X$ THEN 70
   120  BASE(1) = VAL(SUBNO$(3,4))
   130  SUB(1) = 0
   140  PRINT ; INPUT "THE FILE SUBSCRIPT IS (3 #'S 0 - 999) - ", SUB(1)
   150  PRINT "IS ", SUB(1), " CORRECT?"
   160  INPUT X$
   170  IF X$ = "Y" THEN 180 ; IF X$ = X$ THEN 130
   180  IF SUB(1) < 0 OR SUB(1) > 999 THEN 130
   190  SSUB(1) = SUB(1)
   200  PUT 0 IN PDY
   210  PRINT ; INPUT "STIMULATING THE RIGHT OR LEFT WRIST? - ", WRIST$(1)
   220  Y$(W) = WRIST$(1) ; P = 6 ; W = W + P
   230  PRINT ; INPUT "WHAT IS THE STIMULUS DELAY TIME (IN MS)? - ", DELAY(1)
   240  DELAY$(1) = STR$(DELAY(1)) ; Y$(W) = DELAY$(1) ; P = 3 ; W = W + P
   250  PRINT ; PRINT "USER CANNOT REQUEST MORE THAN 13 STIMULI PER SECOND."
   260  INPUT "NUMBER OF STIMULI PER SECOND - ", STSEC(1)
   270  IF STSEC(1) < 1 OR STSEC(1) > 13 THEN 250
   280  STSEC$(1) = STR$(STSEC(1)) ; Y$(W) = STSEC$(1) ; P = 3 ; W = W + P
   290  STPER(1) = 1/STSEC(1) ; REPTS(1) = STPER(1) * 1000
   300  REPTS$(1) = STR$(REPTS(1)) ; Y$(W) = REPTS$(1) ; P = 5 ; W = W + P
   310  PRINT ; INPUT "MINIMUM LENGTH OF THE SWEEP (IN MS) - ", SWPSZ(1)
   320  GOSUB 1510
   330  STMEM$(1) = "0" ; Y$(W) = STMEM$(1) ; P = 2 ; W = W + P
   340  Y$(W) = SZMEM$(1) ; P = 5 ; W = W + P ; Y$(W) = LLEWD$(1) ; P = 4
   350  W = W + P ; Y$(W) = SWPSZ$(1) ; P = 7 ; W = W + P
   360  GOSUB 1620
   370  PUT 2^12 IN LDL
   380  PRINT
   390  INPUT "WHAT IS THE SUBJECT'S THRESHOLD (IN MILLIAMPS)? - ", THRES$(1)
   400  Y$(W) = THRES$(1) ; P = 4 ; W = W + P ; PRINT
   410  INPUT "WHAT IS THE STIMULUS INTENSITY (IN MILLIAMPS)? - ", STINT$(1)
   420  Y$(W) = STINT$(1) ; P = 5 ; W = W + P
   430  PRINT ; INPUT "INPUT THE SENSITIVITY - ", SENTY(1)
   440  IF SENTY(1) = 1.25 OR SENTY(1) = 2.5 OR SENTY(1) = 5 THEN 480
   450  IF SENTY(1) = 12.5 OR SENTY(1) = 25 OR SENTY(1) = 50 THEN 480
   460  IF SENTY(1) = 125 OR SENTY(1) = 250 OR SENTY(1) = 500 THEN 480
   470  GOTO 430
   480  SENTY$(1) = STR$(SENTY(1)) ; Y$(W) = SENTY$(1) ; P = 5 ; W = W + P
```

```
490  PRINT ; INPUT "INPUT THE LOW BANDPASS -", LBAND(1)
500  IF LBAND(1) = 1 OR LBAND(1) = 5 OR LBAND(1) = 30 THEN 530
510  IF LBAND(1) = 150 OR LBAND(1) = 300 THEN 530
520  GOTO 490
530  LBAND$(1) = STR$(LBAND(1)) ; Y$(W) = LBAND$(1) ; P = 4 ; W = W + P
540  PRINT ; INPUT "INPUT THE HIGH BANDPASS - ", HBAND(1)
550  IF HBAND(1) = 30 OR HBAND(1) = 100 OR HBAND(1) = 250 THEN 580
560  IF HBAND(1) = 1500 OR HBAND(1) = 3000 OR HBAND(1) = 8000 THEN 580
570  GOTO 540
580  HBAND$(1) = STR$(HBAND(1)); Y$(W) = HBAND$(1); P = 5 ; W = W + P
590  PRINT ; INPUT "NUMBER OF SWEEPS TO BE COLLECTED - ", SWPS(1)
600  SWPS$(1) = STR$(SWPS(1)) ; Y$(W) = SWPS$(1) ; P = 5 ; W = W + P
610  PUT SWPS(1) IN PSW ; GET SWPS(1) FROM GSW
620  PRINT ; PRINT "USER CAN REQUEST 1, 2, 4, OR 8 CHANNELS TO BE RECORDED."
630  INPUT "NUMBER OF CHANNELS COLLECTED? - ", CHS(1)
640  IF CHS(1) < 1 OR CHS(1) > 8 THEN 620
650  IF CHS(1) = 3 OR CHS(1) = 5 OR CHS(1) = 6 OR CHS(1) = 7 THEN 620
660  PUT CHS(1) IN PIN ; GET CHS(1) FROM GIN ; CHS$(1) = STR$(CHS(1))
670  Y$(W) = CHS$(1) ; P = 2 ; W = W + P ; GOTO 680
680  PRINT ; INPUT "DO YOU NEED A LISTING OF THE PARAMETERS - ", X$
690  IF X$ = "Y" THEN 700 ; IF X$ = "N" THEN 850 ; IF X$ = X$ THEN 680
700  PRINT; PRINT "1.  WRIST STIMULATION - ", WRIST$(1)
710  PRINT "2.  SUBJECTS THRESHOLD - ", THRES$(1)
720  PRINT "3.  STIMULUS INTENSITY - ", STINT$(1)
730  PRINT ; PRINT "4.  STIMULUS DELAY TIME - ", DELAY$(1)
740  PRINT "5.  STIMULUS PER SECOND - ", STSEC$(1)
750  PRINT "6.  LENGTH OF SWEEP - ", SWPSZ$(1)
760  PRINT "7.  DWELL TIME -     ", DWELL$(1)
770  PRINT "8.  STIMULUS PERIOD - ", REPTS$(1)
780  PRINT ; PRINT "9.  NUMBER OF SWEEPS -     ", SWPS$(1)
790  PRINT "10. NUMBER OF CHANNELS -      ", CHS$(1)
800  PRINT "11. START OF MEMORY - ", STMEM$(1)
810  PRINT "12. SIZE OF MEMORY - ", SZMEM$(1)
820  PRINT ; PRINT "13. AMPLIFIER/FILTER SENSITIVITY -     ", SENTY$(1)
830  PRINT "14. AMPLIFIER/FILTER LOW BANDPASS - ", LBAND$(1)
840  PRINT "15. AMPLIFIER/FILTER HIGH BANDPASS - ", HBAND$(1)
850  PRINT
860  INPUT "ARE ANY OF THE ABOVE PARAMETER VALUES INCORRECT - ", X$
870  IF X$ = "Y" THEN 880 ; IF X$ = "N" THEN 1200 ; GOTO 850
880  PRINT ; INPUT "GIVE THE # OF THE PARAMETER TO BE CHANGED - ", NO(1)
890  PRINT "IS ", NO(1), " CORRECT?"
900  INPUT X$
910  IF X$ = "Y" THEN 920 ; IF X$ = X$ THEN 880
920  IF NO(1) = 1 THEN 990 ; IF NO(1) = 2 THEN 1000 ; IF NO(1) = 3 THEN 1010
930  IF NO(1) = 4 THEN 1020 ; IF NO(1) = 5 THEN 1040 ; IF NO(1) = 6 THEN 1080
940  IF NO(1) = 7 OR NO(1) = 8 OR NO(1) = 11 OR NO(1) = 12 THEN 970
950  IF NO(1) = 9 THEN 1130 ; IF NO(1) = 10 THEN 1150 ; IF NO(1) = 13 THEN 1170
960  IF NO(1) = 14 THEN 1180; IF NO(1) = 15 THEN 1190; IF NO(1)=NO(1) THEN 880
970  PRINT ; PRINT "THIS VARIABLE IS NOT UNDER USER CONTROL."
980  GOTO 850
990  N1(1) = 1 ; N2(1) = 5 ; GOSUB 1750 ; WRIST$(1) = VAL$(1) ; GOTO 680
```

```
1000  N1(1) = 36 ; N2(1) = 38 ; GOSUB 1750 ; THRES$(1) = VAL$(1) ; GOTO 680
1010  N1(1) = 40 ; N2(1) = 43 ; GOSUB 1750 ; STINT$(1) = VAL$(1) ; GOTO 680
1020  N1(1) = 8 ; N2(1) = 9 ; GOSUB 1750 ; DELAY$(1) = VAL$(1)
1030  DELAY(1) = VAL(DELAY$(1)) ; GOTO 680
1040  N1(1) = 11 ; N2(1) = 12 ; GOSUB 1750 ; STSEC$(1) = VAL$(1)
1050  STSEC(1) = VAL(STSEC$(1)); STPER(1) = 1/STSEC(1); REPTS(1) = STPER(1)*1000
1060  N1(1) = 13; N2(1) = 14 ; VAL$(1) = STR$(REPTS(1)) ; GOSUB 1790
1070  REPTS$(1) = VAL$(1) ; GOTO 680
1080  N1(1) = 30 ; N2(1) = 34 ; GOSUB 1750 ; SWPSZ(1) = VAL(VAL$(1))
1090  GOSUB 1510
1100  N1(1) = 20 ; N2(1) = 24 ; VAL$(1) = SZMEM$(1) ; GOSUB 1790
1110  N1(1) = 25 ; N2(1) = 28 ; VAL$(1) = LLEWD$(1) ; GOSUB 1790
1120  N1(1) = 29 ; N2(1) = 34 ; VAL$(1) = SWPSZ$(1) ; GOSUB 1790
1130  N1(1) = 60 ; N2(1)= 63 ; GOSUB 1720 ; SWPS$(1) = VAL$(1)
1140  SWPS(1)=VAL(SWPS$(1)); PUT SWPS(1) IN PSW ; GET SWPS(1) FROM GSW ; GOTO 680
1150  N1(1) = 65 ; N2(1) = 65 ; GOSUB 1750 ; CHS$(1) = VAL$(1)
1160  CHS(1)=VAL(CHS$(1)); PUT CHS(1) IN PIN ; GET CHS(1) FROM GIN; GOTO 680
1170  N1(1) = 46 ; N2(1) = 49 ; GOSUB 1750 ; SENTY$(1) = VAL$(1) ; GOTO 680
1180  N1(1) = 51 ; N2(1) = 53 ; GOSUB 1750 ; LBAND$(1) = VAL$(1) ; GOTO 680
1190  N1(1) = 55 ; N2(1) = 58 ; GOSUB 1750 ; HBAND$(1) = VAL$(1) ; GOTO 680
1200  PUT 2 IN PDV ; PUT 208 IN PFM ; PUT BASE(1) IN PFB ; PUT SSUB(1) IN PFS
1210  FOR S = 1 TO W
1220  A$ = Y$(S,S) ; DMEM(S) = ASC(A$)
1230  NEXT S
1240  XEQ DWR
1250  XEQ ZEM
1260  PRINT ; GOSUB 1650
1270  PRINT ; PRINT "HIT ANY KEY TO START AVERAGING WAVEFORMS."
1280  GET X FROM AEN ; PRINT
1290  XEG AVG ; XEQ BSC ; PUT 2 12 IN LDL
1300  PRINT ; INPUT "DO YOU WISH TO STORE THESE WAVEFORMS? - ", X$
1310  IF X$ = "Y" THEN 1320 ; IF X$ = "N" THEN 1330 ; IF X$ = X$ THEN 1300
1320  PUT 196 IN PFM ; XEQ DWR ; SSUB(1) = SSUB(1) + 1
1330  PRINT ; PRINT "DO YOU WANT A PLOT OF THE WAVEFORMS AS THEY APPEAR ON THE"
1340  PRINT "   SCREEN.  NOTE:  NO PARAMETER INFORMATION OR SCALES INCLUDED."
1350  INPUT X$
1360  IF X$ = "Y" THEN 1370 ; IF X$ = "N" THEN 1410 ; IF X$ = X$ THEN 1330
1370  PUT 0 IN PPM ; XEQ PLT
1380  PRINT "WOULD YOU LIKE ANOTHER PLOT OF THE SAME WAVEFORM?  MAKE ANY"
1390  INPUT "  CHANGES BEFORE TYPING A 'Y' RESPONSE.", X$
1400  IF X$ = "Y" THEN 1370 ; IF X$ = X$ THEN 1410
1410  PRINT ; PRINT "IF ANY PARAMETERS ARE TO BE CHANGED REMEMBER TO WRITE A"
1420  PRINT "  NEW PERSONAL FILE."
1430  PRINT ; INPUT "DO YOU WISH TO CONTINUE WITH THIS PROGRAM - ", X$
1440  IF X$ = "N" THEN 1470 ; IF X$ = "Y" THEN 1450 ; IF X$ = X$ THEN 1410
1450  PRINT ; INPUT "DO YOU NEED TO CHANGE ANY PAPAMETER VALUES? - ", X$
1460  IF X$ = "N" THEN 1250 ; IF X$ = "Y" THEN 680 ; IF X$ = X$ THEN 1450
1470  PRINT ; INPUT "DO YOU WISH TO CONTINUE WITH ANOTHER PROGRAM? - ", X$
1480  IF X$ = "Y" THEN 1490 ; IF X$ = "N" THEN 1800 ; IF X$ = X$ THEN 1470
1490  PUT 1 IN PDV ; RUN EPPROG
1500  STOP
```

```
1510  SZMEM(1) = SWPSZ(1) / .04 ; FOR S = 1 TO 14
1520  IF SZMEM(1) < 2^S THEN 1550 ; NEXT S
1530  PRINT ; PRINT "SWEEP LENGTH ERROR - TRY ANOTHER VALUE."
1540  GOTO 310
1550  SZMEM(1) = 2^S
1560  SZMEM$(1) = STR$(SZMEM(1)) ; DWELL(1) = SWPSZ(1) / SZMEM(1) * 1000
1570  IF DWELL(1) < 40 THEN 1580 ; GOTO 1590
1580  DWELL(1) = 40
1590  PUT DWELL(1) IN PDW ; GET DWELL(1) FROM GDW ; LLEWD(1) = DWELL(1) / 1000
1600  LLEWD$(1) = STR$(LLEWD(1)) ; SWPSZ(1) = LLEWD(1) * SZMEM(1)
1610  SWPSZ$(1) = STR$(SWPSZ(1)) ; RETURN
1620  PRINT ; PRINT "FIND SUBJECT'S THRESHOLD AND DETERMINE STIMULUS INTENSITY."
1630  PRINT ; PRINT "EXAMINE THE SIGNAL AND SET THE AMPLIFIER/FILTER'S"
1640  PRINT "   SENSITIVITY AND BANDPASS WHILE IN THE VIEW MODE."
1650  PRINT ; PRINT "TYPE Q WHEN READY TO CONTINUE."
1660  PUT 0 IN PST ; PUT SZMEM(1) IN PSZ
1670  PUT 2^12 IN LDL
1680  PUT 1 IN STN ; PUT REPTS(1) IN LDT
1690  PUT 2 IN STN ; PUT STPER(1) IN LDT
1700  PUT 3 IN STN ; PUT DELAY(1) IN LDT
1710  PUT 2^12 IN CLR
1720  PUT 1 IN ONT
1730  XEQ NTR ; XEQ VIW
1740  RETURN
1750  VAL$(1) = "        " ; PRINT ; INPUT "INPUT THE CORRECT VALUE - ", VAL$(1)
1760  PRINT "IS ", VAL$(1), " CORRECT?"
1770  INPUT X$
1780  IF X$ = "1" THEN 1790 ; IF X$ =X$ THEN 1750
1790  Y$(N1(1),N2(1)) = VAL$(1) ; RETURN
1800  END
$

OK
```

```
LIST DATABU
$
    5  REM ZERO DATA MEMORY
   10  PUT 0 IN PST ; PUT 8192 IN PSZ ; XEQ ZEM
   15  REM DIMENSION VARIABLES AND ASSIGN VALUES TO SOME VARIABLES.
   20  DIM DELAY$(3), STSEC$(3), SWPSZ$(7), SWPS$(5), CHS$(2), STMEM$(2), WRIST$(6)
   30  DIM SZMEM$(5) , DWELL$(5), STPER$(6) , REPTS$(5), Y$(70) , A$(70)
   40  DIM LBAND$(4), HBAND$(5), SENTY$(5), STINT$(5), THRES$(4), SUBNO$(5), VAL$(7)
   50  DIM LLEWD$(5) , X$(1) ; N1(1) = 0 ; N2(1) = 0
   60  W = 1 ; SWPSZ(1) = 0 ; GOTO 70
   65  REM INPUT SUBJECT'S I.D. NUMBER.
   70  SUBNO$(1) = "     "
   80  PRINT ; INPUT "INPUT THE SUBJECT'S I.D. NUMBER - ", SUBNO$(1)
   85  REM VALIDATE VALUE
   90  PRINT "IS ", SUBNO$(1), " CORRECT?"
  100  INPUT X$
  11r  IF X$ = "Y" THEN 120 ; IF X$ = X$ THEN 70
  115  REM FIND FILE BASE AND FILE HEADER.
  120  BASE(1) = VAL(SUBNO$(3,4))
  130  SUB(1) = 0
  135  REM INPUT FILE SUBSCRIPT
  140  PRINT ; INPUT "THE FILE SUBSCRIPT IS (3 #'S 0 - 999) - ", SUB(1)
  145  REM VALIDATE VALUE
  150  PRINT "IS ", SUB(1), " CORRECT?"
  160  INPUT X$
  170  IF X$ = "Y" THEN 180 ; IF X$ = X$ THEN 130
  175  REM CHECK TO INSURE THAT SUBSCRIPT IS IN THE CORRECT RANGE.
  180  IF SUB(1) < 0 OR SUB(1) > 999 THEN 130
  185  REM SET VARIABLE EQUAL TO THE FILE SUBSCRIPT
  190  SSUB(1) = SUB(1)
  195  REM SET DELAY TIME TO ZERO
  200  PUT 0 IN PDY
  205  REM INPUT A VARIABLE'S VALUE AND THE PROGRAM WILL PLACE THAT VALUE INTO
  206  REM   ITS PROPER LOCATION IN THE FILE.
  210  PRINT ; INPUT "STIMULATING THE RIGHT OR LEFT WRIST? - ", WRIST$(1)
  220  Y$(W) = WRIST$(1) ; P = 6 ; W = W + P
  230  PRINT ; INPUT "WHAT IS THE STIMULUS DELAY TIME (IN MS)? - ", DELAY(1)
  240  DELAY$(1) = STR$(DELAY(1)) ; Y$(W) = DELAY$(1) ; P = 3 ; W = W + P
  250  PRINT ; PRINT "USER CANNOT REQUEST MORE THAN 13 STIMULI PER SECOND."
  260  INPUT "NUMBER OF STIMULI PER SECOND - ", STSEC(1)
  265  REM CHECK TO INSURE THAT STIMULI PER SECOND IS IN THE CORRECT DATA RANGE.
  270  IF STSEC(1) < 1 OR STSEC(1) > 13 THEN 250
  280  STSEC$(1) = STR$(STSEC(1)) ; Y$(W) = STSEC$(1) ; P = 3 ; W = W + P
  285  REM MULTIPLY STIMULUS PERIOD BY 1000 SO IT WILL BE AT THE SAME DATA
  286  REM    LEVEL AS THE OTHER DATA.
  290  STPER(1) = 1/STSEC(1) ; REPTS(1) = STPER(1) * 1000
  300  REPTS$(1) = STR$(REPTS(1)) ; Y$(W) = REPTS$(1) ; P = 5 ; W = W + P
  310  PRINT ; INPUT "MINIMUM LENGTH OF THE SWEEP (IN MS) - ", SWPSZ(1)
  320  GOSUB 1510
  325  REM SET START OF DATA MEMORY TO 0.
```

```
330  STMEM$(1) = "0" ; Y$(W) = STMEM$(1) ; P = 2 ; W = W + P
340  Y$(W) = SZMEM$(1) ; P = 5 ; W = W + P ; Y$(W) = LLEWD$(1) ; P = 4
345  REM TURN STIMULUS OFF
350  W = W + P ; Y$(W) = SWPSZ$(1) ; P = 7 ; W = W + P
355  REM TURN STIMULUS ON TO FIND SUBJECT'S THRESHOLD.
360  GOSUB 1620
370  PUT 2^12 IN LDL
380  PRINT
385  REM INPUT A VARIABLE'S VALUE AND THE PROGRAM WILL PLACE THAT VALUE INTO
386  REM   ITS PROPER LOCATION IN THE FILE.
390  INPUT "WHAT IS THE SUBJECT'S THRESHOLD (IN MILLIAMPS)? - ", THRES$(1)
400  Y$(W) = THRES$(1) ; P = 4 ; W = W + P ; PRINT
410  INPUT "WHAT IS THE STIMULUS INTENSITY (IN MILLIAMPS)? - ", STINT$(1)
420  Y$(W) = STINT$(1) ; P = 5 ; W = W + P
430  PRINT ; INPUT "INPUT THE SENSITIVITY - ", SENTY(1)
435  REM CHECK THE VALUE OF SENSITIVITY.
440  IF SENTY(1) = 1.25 OR SENTY(1) = 2.5 OR SENTY(1) = 5 THEN 480
450  IF SENTY(1) = 12.5 OR SENTY(1) = 25 OR SENTY(1) = 50 THEN 480
460  IF SENTY(1) = 125 OR SENTY(1) = 250 OR SENTY(1) = 500 THEN 480
470  GOTO 430
480  SENTY$(1) = STR$(SENTY(1)) ; Y$(W) = SENTY$(1) ; P = 5 ; W = W + P
490  PRINT ; INPUT "INPUT THE LOW BANDPASS -", LBAND(1)
495  REM CHECK THE VALUE OF LOW BANDPASS
500  IF LBAND(1) = 1 OR LBAND(1) = 5 OR LBAND(1) = 30 THEN 530
510  IF LBAND(1) = 150 OR LBAND(1) = 300 THEN 530
520  GOTO 490
530  LBAND$(1) = STR$(LBAND(1)) ; Y$(W) = LBAND$(1) ; P = 4 ; W = W + P
540  PRINT ; INPUT "INPUT THE HIGH BANDPASS - ", HBAND(1)
545  REM CHECK THE VALUE OF HIGH BANDPASS.
550  IF HBAND(1) = 30 OR HBAND(1) = 100 OR HBAND(1) = 250 THEN 580
560  IF HBAND(1) = 1500 OR HBAND(1) = 3000 OR HBAND(1) = 8000 THEN 580
570  GOTO 540
580  HBAND$(1) = STR$(HBAND(1)); Y$(W) = HBAND$(1); P = 5 ; W = W + P
590  PRINT ; INPUT "NUMBER OF SWEEPS TO BE COLLECTED - ", SWPS(1)
600  SWPS$(1) = STR$(SWPS(1)) ; Y$(W) = SWPS$(1) ; P = 5 ; W = W + P
610  PUT SWPS(1) IN PSW ; GET SWPS(1) FROM GSW
620  PRINT ; PRINT "USER CAN REQUEST 1, 2, 4, OR 8 CHANNELS TO BE RECORDED."
630  INPUT "NUMBER OF CHANNELS COLLECTED? - ", CHS(1)
635  REM CHECK THE RANGE OF THE NUMBER OF CHANNELS.
640  IF CHS(1) < 1 OR CHS(1) > 8 THEN 620
650  IF CHS(1) = 3 OR CHS(1) = 5 OR CHS(1) = 6 OR CHS(1) = 7 THEN 620
660  PUT CHS(1) IN PIN ; GET CHS(1) FROM GIN ; CHS$(1) = STR$(CHS(1))
670  Y$(W) = CHS$(1) ; P = 2 ; W = W + P ; GOTO 680
675  REM ENABLES USER TO GET A LISTING OF PARAMETERS AND THEIR VALUES.
680  PRINT ; INPUT "DO YOU NEED A LISTING OF THE PARAMETERS - ", X$
690  IF X$ = "Y" THEN 700 ; IF X$ = "N" THEN 850 ; IF X$ = X$ THEN 680
700  PRINT; PRINT "1.  WRIST STIMULATION - ", WRIST$(1)
710  PRINT "2.  SUBJECTS THRESHOLD - ", THRES$(1)
720  PRINT "3.  STIMULUS INTENSITY - ", STINT$(1)
730  PRINT ; PRINT "4.  STIMULUS DELAY TIME - ", DELAY$(1)
740  PRINT "5.  STIMULUS PER SECOND - ", STSEC$(1)
```

```
750  PRINT "6.  LENGTH OF SWEEP - ", SWPSZ$(1)
760  PRINT "7.  DWELL TIME -    ", LLEWD$(1)
770  PRINT "8.  STIMULUS PERIOD - ", REPTS$(1)
780  PRINT ; PRINT "9.  NUMBER OF SWEEPS -     ", SWPS$(1)
790  PRINT "10. NUMBER OF CHANNELS -     ", CHS$(1)
800  PRINT "11. START OF MEMORY - ", STMEM$(1)
810  PRINT "12. SIZE OF MEMORY - ", SZMEM$(1)
820  PRINT ; PRINT "13. AMPLIFIER/FILTER SENSITIVITY -    ", SENTY$(1)
830  PRINT "14. AMPLIFIER/FILTER LOW BANDPASS - ", LBAND$(1)
840  PRINT "15. AMPLIFIER/FILTER HIGH BANDPASS - ", HBAND$(1)
850  PRINT
855  REM ENABLES USER TO CHANGE THE VALUE OF A VARIABLE.
860  INPUT "ARE ANY OF THE ABOVE PARAMETER VALUES INCORRECT - ", X$
870  IF X$ = "Y" THEN 880 ; IF X$ = "N" THEN 1200 ; GOTO 850
880  PRINT ; INPUT "GIVE THE # OF THE PARAMETER TO BE CHANGED - ", NO(1)
890  PRINT "IS ", NO(1), " CORRECT?"
900  INPUT X$
910  IF X$ = "Y" THEN 920 ; IF X$ = X$ THEN 880
920  IF NO(1) = 1 THEN 990 ; IF NO(1) = 2 THEN 1000 ; IF NO(1) = 3 THEN 1010
930  IF NO(1) = 4 THEN 1020 ; IF NO(1) = 5 THEN 1040 ; IF NO(1) = 6 THEN 1080
940  IF NO(1) = 7 OR NO(1) = 8 OR NO(1) = 11 OR NO(1) = 12 THEN 970
950  IF NO(1) = 9 THEN 1130 ; IF NO(1) = 10 THEN 1150 ; IF NO(1) = 13 THEN 1170
960  IF NO(1) = 14 THEN 1180; IF NO(1) = 15 THEN 1190; IF NO(1)=NO(1) THEN 880
970  PRINT ; PRINT "THIS VARIABLE IS NOT UNDER USER CONTROL."
980  GOTO 850
990  N1(1) = 1 ; N2(1) = 5 ; GOSUB 1750 ; WRIST$(1) = VAL$(1) ; GOTO 680
1000 N1(1) = 36 ; N2(1) = 38 ; GOSUB 1750 ; THRES$(1) = VAL$(1) ; GOTO 680
1010 N1(1) = 40 ; N2(1) = 43 ; GOSUB 1750 ; STINT$(1) = VAL$(1) ; GOTO 680
1020 N1(1) = 8 ; N2(1) = 9 ; GOSUB 1750 ; DELAY$(1) = VAL$(1)
1030 DELAY(1) = VAL(DELAY$(1)) ; GOTO 680
1040 N1(1) = 11 ; N2(1) = 12 ; GOSUB 1750 ; STSEC$(1) = VAL$(1)
1050 STSEC(1) = VAL(STSEC$(1)); STPER(1) = 1/STSEC(1); REPTS(1) = STPER(1)*1000
1060 N1(1) = 13; N2(1) = 14 ; VAL$(1) = STR$(REPTS(1)) ; GOSUB 1790
1070 REPTS$(1) = VAL$(1) ; GOTO 680
1080 N1(1) = 30 ; N2(1) = 34 ; GOSUB 1750 ; SWPSZ(1) = VAL(VAL$(1))
1090 GOSUB 1510
1100 N1(1) = 20 ; N2(1) = 24 ; VAL$(1) = SZMEM$(1) ; GOSUB 1790
1110 N1(1) = 25 ; N2(1) = 28 ; VAL$(1) = LLEWD$(1) ; GOSUB 1790
1120 N1(1) = 29 ; N2(1) = 34 ; VAL$(1) = SWPSZ$(1) ; GOSUB 1790
1130 N1(1) = 60 ; N2(1)= 63 ; GOSUB 1720 ; SWPS$(1) = VAL$(1)
1140 SWPS(1)=VAL(SWPS$(1)); PUT SWPS(1) IN PSW ; GET SWPS(1) FROM GSW ; GOTO 680
1150 N1(1) = 65 ; N2(1) = 65 ; GOSUB 1750 ; CHS$(1) = VAL$(1)
1160 CHS(1)=VAL(CHS$(1)); PUT CHS(1) IN PIN ; GET CHS(1) FROM GIN; GOTO 680
1170 N1(1) = 46 ; N2(1) = 49 ; GOSUB 1750 ; SENTY$(1) = VAL$(1) ; GOTO 680
1180 N1(1) = 51 ; N2(1) = 53 ; GOSUB 1750 ; LBAND$(1) = VAL$(1) ; GOTO 680
1190 N1(1) = 55 ; N2(1) = 58 ; GOSUB 1750 ; HBAND$(1) = VAL$(1) ; GOTO 680
1195 REM SELECT DISK DRIVE # 2 AND THE FILENAME FOR THE PARAMETER FILE.
1200 PUT 2 IN PDV ; PUT 208 IN PFN ; PUT BASE(1) IN PFB ; PUT SSUB(1) IN PFS
1205 REM TRANSULATE PARAMETER DATA FILE INTO ASCII CODE.
1210 FOR S = 1 TO W
1220 A$ = Y$(S,S) ; DMEM(S) = ASC(A$)
```

```
1230  NEXT S
1235  REM SAVE THE PARAMETER DATA FILE ON DISK.
1240  XEQ DWR
1245  REM CLEAR MEMORY.
1250  XEQ ZEM
1255  REM TURN STIMULUS ON.
1260  PRINT ; GOSUB 1650
1270  PRINT ; PRINT "HIT ANY KEY TO START AVERAGING WAVEFORMS."
1280  GET X FROM AEN ; PRINT
1285  REM AVERAGE E.P.'S, BASELINE ZERO, AND TURN STIMULUS OFF.
1290  XEQ AVG ; XEQ BSC ; PUT 2^12 IN LDL
1295  REM ENABLES USER TO STORE WAVEFORMS ON DISK.
1300  PRINT ; INPUT "DO YOU WISH TO STORE THESE WAVEFORMS? - ", X$
1310  IF X$ = "Y" THEN 1320 ; IF X$ = "N" THEN 1330 ; IF X$ = X$ THEN 1300
1315  REM STORE WAVEFORMS AND INCRIMENT THE SUBSCRIPT BY 1.
132(  PUT 196 IN PFH ; XEQ DWR ; SSUB(1) = SSUB(1) + 1
1325  REM ENABLES USER TO PLOT EVOKED POTENTIALS.
1330  PRINT ; PRINT "DO YOU WANT A PLOT OF THE WAVEFORMS AS THEY APPEAR ON THE"
1340  PRINT "   SCREEN.  NOTE:  NO PARAMETER INFORMATION OR SCALES INCLUDED."
1350  INPUT X$
1360  IF X$ = "Y" THEN 1370 ; IF X$ = "N" THEN 1410 ; IF X$ = X$ THEN 1330
1370  PUT 0 IN PPM ; XEQ PLT
1375  REM ENABLES USER TO GET A SECOND PLOT OF THE EVOKED POTENTIALS.
1380  PRINT "WOULD YOU LIKE ANOTHER PLOT OF THE SAME WAVEFORM?  MAKE ANY"
1390  INPUT "  CHANGES BEFORE TYPING A 'Y' RESPONSE.", X$
1400  IF X$ = "Y" THEN 1370 ; IF X$ = X$ THEN 1410
1410  PRINT ; PRINT "IF ANY PARAMETERS ARE TO BE CHANGED REMEMBER TO WRITE A"
1420  PRINT "  NEW PERSONAL FILE."
1425  REM ENABLES USER TO RERUN PROGRAM.
1430  PRINT ; INPUT "DO YOU WISH TO CONTINUE WITH THIS PROGRAM - ", X$
1440  IF X$ = "N" THEN 1470 ; IF X$ = "Y" THEN 1450 ; IF X$ = X$ THEN 1410
1445  REM RERUN THE PROGRAM BUT CHANGE SOME PARAMETER DATA
1450  PRINT ; INPUT "DO YOU NEED TO CHANGE ANY PAPAMETER VALUES? - ", X$
1460  IF X$ = "N" THEN 1250 ; IF X$ = "Y" THEN 680 ; IF X$ = X$ THEN 1450
1465  REM ENABLES USER TO RUN ANOTHER PROGRAM.
1470  PRINT ; INPUT "DO YOU WISH TO CONTINUE WITH ANOTHER PROGRAM? - ", X$
1480  IF X$ = "Y" THEN 1490 ; IF X$ = "N" THEN 1800 ; IF X$ = X$ THEN 1470
1490  PUT 1 IN PDV ; RUN EPPROG
1495  REM STOP MAIN PROGRAM, THE FOLLOWING ARE SUBROUTINES.
1500  STOP
1505  REM SUBROUTINE TO DETERMINE VALUE OF SWEEPLENGTH, SIZE OF DMEM, AND DWELL.
1510  SZMEM(1) = SWPSZ(1) / .04 ; FOR S = 1 TO 14
1515  REM INSURES THAT THE SIZE OF DMEM IS A FACTOR OF 2.
1520  IF SZMEM(1) < 2^S THEN 1550 ; NEXT S
1530  PRINT ; PRINT "SWEEP LENGTH ERROR - TRY ANOTHER VALUE."
1540  GOTO 310
1550  SZMEM(1) = 2^S
1560  SZMEM$(1) = STR$(SZMEM(1)) ; DWELL(1) = SWPSZ(1) / SZMEM(1) * 1000
1565  REM IF DWELL IS LESS THAN 40 SET DWELL EQUAL TO 40.
1570  IF DWELL(1) < 40 THEN 1580 ; GOTO 1590
1580  DWELL(1) = 40
```

```
1590  PUT DWELL(1) IN PDW ; GET DWELL(1) FROM GDW ; LLEWD(1) = DWELL(1) / 1000
1600  LLEWD$(1) = STR$(LLEWD(1)) ; SWPSZ(1) = LLEWD(1) * SZMEM(1)
1610  SWPSZ$(1) = STR$(SWPSZ(1)) ; RETURN
1615  REM SUBROUTINE TO TURN ON STIMULUS.
1620  PRINT ; PRINT "FIND SUBJECT´S THRESHOLD AND DETERMINE STIMULUS INTENSITY."
1630  PRINT ; PRINT "EXAMINE THE SIGNAL AND SET THE AMPLIFIER/FILTER´S"
1640  PRINT "   SENSITIVITY AND BANDPASS WHILE IN THE VIEW MODE."
1650  PRINT ; PRINT "TYPE Q WHEN READY TO CONTINUE."
1660  PUT 0 IN PST ; PUT SZMEM(1) IN PSZ
1670  PUT 2^12 IN LDL
1675  REM INPUT PARAMETERS FOR THE STIMULUS THROUGH NIC-293.
1680  PUT 1 IN STN ; PUT REPTS(1) IN LDT
1690  PUT 2 IN STN ; PUT STPER(1) IN LDT
1700  PUT 3 IN STN ; PUT DELAY(1) IN LDT
1705  REM TURN THE STIMULUS ON.
1710  PUT 2^12 IN CLR
1720  PUT 1 IN ONT
1725  REM SUBROUTINE TO CHANGE A VARIABLE´S VALUE.
1730  XEQ NTR ; XEQ VIW
1740  RETURN
1750  VAL$(1) = "       " ; PRINT ; INPUT "INPUT THE CORRECT VALUE - ", VAL$(1)
1760  PRINT "IS ", VAL$(1), " CORRECT?"
1770  INPUT X$
1780  IF X$ = "Y" THEN 1790 ; IF X$ =X$ THEN 1750
1790  Y$(N1(1),N2(1)) = VAL$(1) ; RETURN
1795  REM END OF THE PROGRAM.
1800  END
$

OK
```

```
LIST PLOTEP
$
  10  DIM A$(9),B$(6),C$(6),D$(16),E$(17),F$(30),G$(10),H$(7),I$(20),J$(16),K$(20)
  20  DIM L$(18),M$(1),O$(1),W$(18),CH$(5),V$(225),C1$(5),C2$(5),C3$(5),C4$(5)
  30  DIM C5$(5), C6$(5), C7$(5), C8$(5), Y$(66), SUBNO$(4), SUB(3), GA$(11)
  40  DIM GAIN$(6),NO$(15),FN$(10),SUB$(4),NOF$(7) ; Y$=" ";V$=" ";A$="G-FORCE: "
  50  B$="DATE: ";C$="NAME: ";D$="SUBJECT NUMBER: ";G$="(IN MSEC.)";H$="(IN UV)"
  60  E$="SLED RUN NUMBER: ";F$="SOMATOSENSORY EVOKEU POTENTIAL";I$="DELAY TIME: "
  70  J$="STIMULI/SECOND: ";K$="DWELL TIME (IN US): ";L$="SWEEP LENGTH: ";M$="-"
  80  GA$="PLOT GAIN: ";NO$="NUMBER SWEEPS: ";FN$="FILENAME: ";M=-.3;C=-.9;CH(1)=0
  90  W$ = "WRIST STIMULATED: "; C1$ = "CH.#1" ; C2$ = "CH.#2" ; C3$ = "CH.#3"
 100  C4$ = "CH.#4" ; C5$ = "CH.#5" ; C6$ = "CH.#6" ; C7$ = "CH.#7" ; C8$ = "CH.#8"
 110  X(1) = 0 ; D(1) = 0 ; N1(1) = 0 ; N2(1) = 0 ; N3(1) = 0 ; N4(1) = 0
 120  PUT 0 IN PST ; PUT 1024 IN PSZ ; PUT 2 IN PDV
 130  PRINT ; INPUT "INPUT THE SUBJECT'S I.D. NUMBER - ", SUBNO$(1)
 140  PRINT "IS ", SUBNO$(1), " CORRECT?"
 150  INPUT X$
 160  IF X$ = "Y" THEN 170 ; GOTO 130
 170  BASE(1) = VAL(SUBNO$(3,4)) ; HEAD(1) = ASC(SUBNO$(1))
 180  PRINT ; INPUT "INPUT THE SUBSCRIPT NUMBER (3 #'S 0 - 999)? - ", SUB(1)
 190  PRINT "IS ", SUB(1), " CORRECT?"
 200  INPUT X$
 210  IF X$ = "Y" THEN 220 ; GOTO 180
 220  IF SUB(1) < 0 OR SUB(1) > 999 THEN 180
 230  IF HEAD(1) = 40 THEN 240 ; IF HEAD(1) = 50 THEN 250
 240  PUT 200 IN PFH ; PUT BASE(1) IN PFB ; PUT SUB(1) IN PFS ; GOTO 260
 250  PUT 210 IN PFH ; PUT BASE(1) IN PFB ; PUT SUB(1) IN PFS
 260  XEQ DRD
 270  FOR S = 1 TO 225
 280  U = DMEM(S) ; V$(S,S) = CHR$(U) ; NEXT S
 290  PUT 208 IN PFH ; XEQ DRD
 300  FOR S = 1 TO 66
 310  UU = DMEM(S) ; Y$(S,S) = CHR$(UU) ; NEXT S
 320  HS(1)=VAL(Y$(30,34)); O$(1)=V$(143); CH(1)=VAL(O$(1)); HR(1) = HS(1) / 10
 330  PRINT ; INPUT "WHAT IS THE MIN. VOLTAGE LEVEL (A NEG. #) - ", VA(1)
 340  PRINT "IS ", VA(1), " CORRECT?"
 350  INPUT X$
 360  IF X$ = "Y" THEN 370 ; GOTO 330
 370  VR(1) = -VA(1) * 2 / 10
 380  INPUT "INPUT GAIN FOR THE PLOTTING OF E.P. WAVEFORMS - ", GAIN$(1)
 390  PRINT "IS ", GAIN$(1), " CORRECT?"
 400  INPUT X$
 410  IF X$ = "Y" THEN 420 ; GOTO 380
 420  XEQ DEF ; PUT 1.0 IN ROT
 430  SUB$(1) = STR$(SUB(1))
 440  NOF$(1) = SUBNO$(1,1) ; NOF$(2,3) = SUBNO$(3,4) ; NOF$(4,6) = SUB$(2,4)
 450  IF O$(1) = "1" THEN 490 ; IF O$(1) = "2" THEN 720
 460  IF O$(1) = "3" THEN 510 ; IF O$(1) = "4" THEN 680
 470  IF O$(1) = "5" THEN 540 ; IF O$(1) = "6" THEN 640
 480  IF O$(1) = "7" THEN 570 ; IF O$(1) = "8" THEN 600
```

```
490  D(1)=.2; CH$(1)=C1$; N1(1)=215; N2(1)=219; N3(1)=220; N4(1)=224
500  GOSUB 1340 ; GOSUB 1430 ; GOSUB 1390 ; GOTO 750
510  SIZE 0,-8.5 ; VECT PENUP, 1,-1 ; XEQ DEF ; PUT 1.0 IN ROT
520  D(1)=.2; CH$(1)=C3$; N1(1)=195; N2(1)=199; N3(1)=200; N4(1)=204
530  GOSUB 1340 ; GOSUB 1430 ; GOSUB 1390 ; GOTO 810
540  SIZE 0,-17 ; VECT PENUP, 1,-1 ; XEQ DEF
550  D(1)=.2; CH$(1)=C5$; N1(1)=175; N2(1)=179; N3(1)=180; N4(1)=184
560  GOSUB 1340 ; GOSUB 1430 ; GOSUB 1390 ; GOTO 690
570  SIZE 0,-25.5 ; VECT PENUP, 1,-1 ; XEQ DEF
580  D(1)=.2; CH$(1)=C7$; N1(1)=155; N2(1)=159; N3(1)=160; N4(1)=164
590  GOSUB 1340 ; GOSUB 1430 ; GOSUB 1390 ; GOTO 650
600  SIZE 0,-25.5 ; VECT PENUP, 1,-1 ; XEQ DEF
610  D(1)=.2; CH$(1)=C6$; N1(1)=165; N2(1)=169; N3(1)=170; N4(1)=174; GOSUB 1340
620  D(1)=3.85; CH$(1)=C7$; N1(1)=155; N2(1)=159; N3(1)=160; N4(1)=164
630  GOSUB 1340 ; GOSUB 1470 ; GOSUB 1930 ; GOSUB 1390 ; GOTO 650
640  SIZE 0,-17 ; VECT PENUP, 1,-1 ; XEQ DEF
650  D(1)=.2; CH$(1)=C6$; N1(1)=165; N2(1)=169; N3(1)=170; N4(1)=174; GOSUB 1340
660  D(1)=3.85; CH$(1)=C5$; N1(1)=175; N2(1)=179; N3(1)=180; N4(1)=184
670  GOSUB 1340 ; GOSUB 1470 ; GOSUB 1930 ; GOSUB 1390 ; GOTO 690
680  SIZE 0,-8.5 ; VECT PENUP, 1,-1 ; XEQ DEF ; PUT 1.0 IN ROT
690  D(1)=.2; CH$(1)=C4$; N1(1)=185; N2(1)=189; N3(1)=190; N4(1)=194; GOSUB 1340
700  D(1)=3.85; CH$(1)=C3$; N1(1)=195; N2(1)=199; N3(1)=200; N4(1)=204
710  GOSUB 1340 ; GOSUB 1470 ; GOSUB 1930 ; GOSUB 1390
720  D(1)=.2; CH$(1)=C2$; N1(1)=205; N2(1)=209; N3(1)=210; N4(1)=214; GOSUB 1340
730  D(1)=3.85; CH$(1)=C1$; N1(1)=215; N2(1)=219; N3(1)=220; N4(1)=225
740  GOSUB 1340 ; GOSUB 1470 ; GOSUB 1930 ; GOSUB 1390
750  SZ(1) = VAL(Y$(21,24)) ; PLTSZ(1) = SZ(1) / CH(1)
760  PUT 0 IN PST ; PUT SZ(1) IN PSZ ; PUT 2 IN PDV
770  PUT 196 IN PFH ; PUT BASE(1) IN PFB ; PUT SUB(1) IN PFS ; XEQ DRD
780  PUT PLTSZ(1) IN PSZ ; XEQ DEF ; PUT 1.0 IN ROT
790  IF O$(1)="1" THEN 1130; IF O$(1)="2" THEN 1090; IF O$(1)="3" THEN 1040
800  IF O$(1)="4" THEN 1000; IF O$(1)="5" THEN 950; IF O$(1)="6" THEN 910
810  IF O$(1) = "7" THEN 860 ; IF O$(1) = "8" THEN 820
820  SIZE 0,-34 ; VECT PENUP, 1,-1 ; XEQ DEF
830  SIZE -1,-.8 ; VECT PENUP, 1,1
840  PUT PLTSZ(1) * 7 IN PST ; SIZE -9,-3 ; XEQ DPL
850  GOTO 880
860  SIZE 0,-34 ; VECT PENUP, 1,-1 ; XEQ DEF
870  SIZE -1,-.8 ; VECT PENUP, 1,1 ; GOTO 890
880  SIZE -1,-4 ; VECT PENUP, 1,1
890  PUT PLTSZ(1) * 6 IN PST ; SIZE -9,-3 ; XEQ DPL
900  SIZE 0,-8.5 ; VECT PENUP, 1,1 ; GOTO 920
910  SIZE 0,-25.5 ; VECT PENUP, 1,-1
920  XEQ DEF ; SIZE -1,-.8 ; VECT PENUP, 1,1
930  PUT PLTSZ(1) * 5 IN PST ; SIZE -9,-3 ; XEQ DPL
940  GOTO 970
950  SIZE 0,-25.5 ; VECT PENUP, 1,-1 ; XEQ DEF
960  SIZE -1,-.8 ; VECT PENUP, 1,1 ; GOTO 980
970  SIZE -1,-4 ; VECT PENUP, 1,1
980  PUT PLTSZ(1) * 4 IN PST ; SIZE -9,-3 ; XEQ DPL
990  SIZE 0,-8.5 ; VECT PENUP, 1,1 ; GOTO 1010
```

```
1000  SIZE 0,-17 ; VECT PENUP, 1,-1
1010  XEQ DEF ; SIZE -1,-.8 ; VECT PENUP, 1,1
1020  PUT PLTSZ(1)* 3 IN PST ; SIZE -9,-3 ; XEQ DPL
1030  GOTO 1060
1040  SIZE 0,-17 ; VECT PENUP, 1,-1 ; XEQ DEF
1050  SIZE -1,-.8 ; VECT PENUP, 1,1 ; GOTO 1070
1060  SIZE -1,-4 ; VECT PENUP, 1,1
1070  PUT PLTSZ(1) * 2 IN PST ; SIZE -9,-3 ; XEQ DPL
1080  SIZE 0,-8.5 ; VECT PENUP, 1,1 ; GOTO 1100
1090  SIZE 0,-8.5 ; VECT PENUP, 1,-1
1100  XEQ DEF ; SIZE -1,-.8 ; VECT PENUP, 1,1
1110  PUT PLTSZ(1) IN PST ; SIZE -9,-3 ; XEQ DPL
1120  GOTO 1150
1130  SIZE 0,-8.5 ; VECT PENUP, 1,-1 ; XEQ DEF
1140  SIZE -1,-.8 ; VECT PENUP, 1,1 ; GOTO 1160
1150  SIZE -1,-4 ; VECT PENUP, 1,1
1160  PUT 0 IN PST ; SIZE -9,-3 ; XEQ DPL
1170  VECT PENUP, 0,0
1180  PRINT ; INPUT "DO YOU WISH TO PLOT ANOTHER FILE? - ", X$
1190  IF X$ = "Y" THEN 1210 ; IF X$ = "N" THEN 1310 ; GOTO 1180
1200  PRINT ; INPUT "IS THE FILE IN THIS SERIES - ", X$
1210  IF X$ = "Y" THEN 1230 ; IF X$ = "N" THEN 1220 ; GOTO 1200
1220  GOSUB 2100 ; GOTO 120
1230  INPUT "IS THE FILE NEXT IN THIS SEQUENCE? - ", X$
1240  IF X$ = "Y" THEN 1250 ; IF X$ = "N" THEN 1260 ; GOTO 1230
1250  SUB(1) = SUB(1) + 1 ; GOSUB 2100 ; GOTO 420
1260  INPUT "INPUT THE FULL SUBSCRIPT OF THE NEW FILE - ", SUB(1)
1270  PRINT "IS ", SUB(1), " CORRECT?"
1280  INPUT X$
1290  IF X$ = "Y" THEN 1300 ; GOTO 1260
1300  GOSUB 2100 ; GOTO 420
1310  PRINT ; INPUT "DO YOU WANT TO CONTINUE WITH ANOTHER PROGRAM? - ", X$
1320  IF X$ = "Y" THEN 1330 ; IF X$ = "N" THEN 2190 ; GOTO 1210
1330  RUN EPPROG
1335  STOP
1340  SIZE -1.2,-D(1) ; VECT PENUP, 1,1 ; PLOT -.1, CH$(1)
1350  SIZE -2.0,-D(1) ; VECT PENUP, 1,1 ; PLOT -.1, V$(N1(1),N2(1))
1360  SIZE -2.4,-D(1) ; VECT PENUP, 1,1 ; PLOT -.1, M$
1370  SIZE -2.6,-D(1) ; VECT PENUP, 1,1 ; PLOT -.1, V$(N3(1),N4(1))
1380  RETURN ; STOP
1390  VECT PENUP, 0,0 ; PUT 2.0 IN ROT ; SIZE -3,-.2 ; VECT PENUP, 1,-1
1400  PLOT -.1, M$
1410  SIZE -8.5,0 ; VECT PENUP, 1,-1 ; XEQ DEF ; PUT 1.0 IN ROT
1420  RETURN ; STOP
1430  XX = -10 ; SIZE -1,-3.8 ; VECT PENUP, 1,1
1440  SIZE XX,-3.8 ; VECT PENDN, 1,1 ; FOR S = 0 TO 10
1450  SIZE XX,-3.7 ; VECT PENUP, 1,1 ; SIZE XX,-3.9 ; VECT PENDN, 1,1
1460  XX = XX - C ; NEXT S
1470  SIZE -2.4,-8.2 ; VECT PENUP, 1,1 ; PLOT -.2, F$
1480  SIZE -.5,-7.9 ; VECT PENUP, 1,1 ; PLOT -.1, C$
1490  SIZE -1.2,-7.9 ; VECT PENUP, 1,1 ; PLOT -.1, V$(1,34)
```

```
1500  SIZE -.5,-7.7 ; VECT PENUP, 1,1 ; PLOT -.1, FN$
1510  SIZE -1.7,-7.7 ; VECT PENUP, 1,1 ; PLOT -.1, NOF$(1)
1520  SIZE -.5,-7.5 ; VECT PENUP, 1,1 ; PLOT -.1, D$
1530  SIZE -2.2,-7.5 ; VECT PENUP, 1,1 ; PLOT -.1, V$(35,39)
1540  SIZE -.5,-7.3 ; VECT PENUP, 1,1 ; PLOT -.1, E$
1550  SIZE -2.3,-7.3 ; VECT PENUP, 1,1 ; PLOT -.1, V$(56,61)
1560  SIZE -4.75,-7.9 ; VECT PENUP, 1,1 ; PLOT -.1, A$
1570  SIZE -5.7,-7.9 ; VECT PENUP, 1,1 ; PLOT -.1, V$(50,55)
1580  SIZE -4.75,-7.7 ; VECT PENUP, 1,1 ; PLOT -.1, B$
1590  SIZE -5.4,-7.7 ; VECT PENUP, 1,1 ; PLOT -.1, V$(41,48)
1600  SIZE -4.75,-7.5 ; VECT PENUP, 1,1 ; PLOT -.1, W$
1610  SIZE -6.6,-7.5 ; VECT PENUP, 1,1 ; PLOT -.1, Y$(1,5)
1620  SIZE -4.75,-7.3 ; VECT PENUP, 1,1 ; PLOT -.1, I$
1630  SIZE -5.95,-7.3 ; VECT PENUP, 1,1 ; PLOT -.1, Y$(8,9)
1640  SIZE -8,-7.9 ; VECT PENUP, 1,1 ; PLOT -.1, J$
1650  SIZE -9.6,-7.9 ; VECT PENUP, 1,1 ; PLOT -.1, Y$(11,12)
1660  SIZE -8,-7.7 ; VECT PENUP, 1,1 ; PLOT -.1, K$
1670  SIZE -10,-7.7 ; VECT PENUP, 1,1 ; PLOT -.1, Y$(25,28)
1680  SIZE -8,-7.5 ; VECT PENUP, 1,1 ; PLOT -.1, L$
1690  SIZE -9.5,-7.5 ; VECT PENUP, 1,1 ; PLOT -.1, Y$(30,34)
1700  SIZE -8,-7.3 ; VECT PENUP, 1,1 ; PLOT -.1, NO$
1710  SIZE -9.6,-7.3 ; VECT PENUP, 1,1 ; PLOT -.1, Y$(60,63)
1720  SIZE -7.5,-3.85 ; VECT PENUP, 1,1 ; PLOT -.1, GA$
1730  SIZE -8.6,-3.85 ; VECT PENUP, 1,1 ; PLOT -.1, GAIN$(1)
1740  SIZE -10,-.8 ; VECT PENUP, 1,1 ; SIZE -1,-.8 ; VECT PENDN, 1,1
1750  XX = -1 ; UHA(1) = 0 ; FOR S = 0 TO 10
1760  SIZE XX,-.7 ; VECT PENUP, 1,1 ; SIZE XX,-.9 ; VECT PENDN, 1,1
1770  SIZE (XX+.4),-.4 ; VECT PENUP, 1,1 ; PLOT -.1, UHA(1) IN 4,4
1780  XX = XX + C ; UHA(1) = UHA(1) + HR(1) ; NEXT S
1790  SIZE -5,-.2 ; VECT PENUP, 1,1 ; PLOT -.1, G$
1800  SIZE -1,-3.8 ; VECT PENUP, 1,1 ; SIZE -1,-.8 ; VECT PENDN, 1,1
1810  XX = -.8 ; UVA(1) = VA(1) ; FOR S = 0 TO 10
1820  SIZE -.9,XX ; VECT PENUP, 1,1 ; SIZE -1.1,XX ; VECT PENDN, 1,1
1830  SIZE -.35,XX ; VECT PENUP, 1,1 ; PLOT -.1, UVA(1) IN 2,3
1840  XX = XX + M ; UVA(1) = UVA(1) + VR(1) ; NEXT S
1850  SIZE -1,-2.3 ; VECT PENUP, 1,1 ; P = -.1 ; Z = -1.1 ; FOR S = 1 TO 45
1860  SIZE Z,-2.3 ; VECT PENDN, 1,1 ; Z = Z + P
1870  SIZE Z,-2.3 ; VECT PENUP, 1,1 ; Z = Z + P ; NEXT S
1880  SIZE -10,-.8 ; VECT PENUP, 1,1 ; SIZE -10,-3.8 ; VECT PENDN, 1,1
1890  XX = -3.8 ; FOR S = 0 TO 10
1900  SIZE -9.9,XX ; VECT PENUP, 1,1 ; SIZE -10.1,XX ; VECT PENDN, 1,1
1910  XX = XX - M ; NEXT S
1920  RETURN ; STOP
1930  SIZE -1,-4 ; VECT PENUP, 1,1 ; SIZE -1,-7 ; VECT PENDN, 1,1
1940  XX = -4 ; UVA(1) = VA(1) ; FOR S = 0 TO 10
1950  SIZE -.9,XX ; VECT PENUP, 1,1 ; SIZE -1.1,XX ; VECT PENDN, 1,1
1960  SIZE -.35,XX ; VECT PENUP, 1,1 ; PLOT -.1, UVA(1) IN 2,3
1970  XX = XX + M ; UVA(1) = UVA(1) + VR(1) ; NEXT S
1980  XX = -10 ; SIZE -1,-7 ; VECT PENUP, 1,1
1990  SIZE XX,-7 ; VECT PENDN, 1,1 ; FOR S = 0 TO 10
2000  SIZE XX,-6.9 ; VECT PENUP, 1,1 ; SIZE XX,-7.1 ; VECT PENDN, 1,1
```

```
  2010  XX = XX - C ; NEXT S
  2020  SIZE -1,-5.5 ; VECT PENUP, 1,1 ; P = -.1 ; Z = -1.1 ; FOR S = 1 TO 45
  2030  SIZE Z,-5.5 ; VECT PENDN, 1,1 ; Z = Z + P
  2040  SIZE Z,-5.5 ; VECT PENUP, 1,1 ; Z = Z + P ; NEXT S
  2050  SIZE -10,-4 ; VECT PENUP, 1,1 ; SIZE -10,-7 ; VECT PENDN, 1,1
  2060  XX = -7 ; FOR S = 0 TO 10
  2070  SIZE -9.9,XX ; VECT PENUP, 1,1 ; SIZE -10.1,XX ; VECT PENDN, 1,1
  2080  XX = XX - M ; NEXT S
  2090  RETURN ; STOP
  2100  IF O$(1) = "1" OR O$(1) = "2" THEN 2140
  2110  IF O$(1) = "3" OR O$(1) = "4" THEN 2150
  2120  IF O$(1) = "5" OR O$(1) = "6" THEN 2160
  2130  IF O$(1) = "7" OR O$(1) = "8" THEN 2170
  2140  SIZE 0,-8.5 ; VECT PENUP, 1,1 ; GOTO 2180
  2150  SIZE 0,-17 ; VECT PENUP, 1,1 ; GOTO 2180
  2160  SIZE 0,-25.5 ; VECT PENUP, 1,1 ; GOTO 2180
  2170  SIZE 0,-34 ; VECT PENUP, 1,1
  2180  RETURN ; STOP
  2190  END
$

OK
```

```
LIST PLOTBU
$
   5  REM (10 - 110) - DIMENSION VARIABLES AND ASSIGN VALUES TO SOME VARIABLES
 115  REM (120) - SELECT START AND SIZE OF DMEM AND THE DISK DRIVE TO BE USED.
 116  REM   ALSO REPLACES FILES BACK INTO THE SAME LOCATION AS FOUND.
 125  REM (130 - 160) - INPUT THE I.D. NUMBER OF THE SUBJECTS WHOSE FILE IS
 126  REM    TO BE PLOTTED.  USER IS ASKED TO VALIDATE VARIABLE'S VALUE.
 165  REM (170) - THE FILE BASE IS THE LAST TWO #'S OF THE SUBJECT'S I.D. NUMBER
 166  REM   THE FILE HEADER IS THE FIRST DIGIT OF THE SUBJECT'S I.D. NUMBER.
 175  REM (180 - 220) - INPUT THE FILE SUBSCRIPT.  USER IS ASKED TO VALIDATE
 176  REM   VARIABLE'S VALUE.  FOLLOWED BY A CHECK ON VALUES RANGE.
 225  REM (230 - 250) - THE FILE HEADER IS CHECKED TO DETERMINE IF THE SUBJECT
 226  REM   WAS HUMAN OR A RHESUS.  THE CORRECT RESPONSE IS THEN PLACED INTO THE
 227  REM   FILE HEADER.  THE BASE AND SUBSCRIPT NUMBER ARE ALSO SET.
 255  REM (260) - THE FILE SECLECTED IS READ FROM DISK TO DMEM.
 265  REM (270 - 280) - THE SUBJECT'S PERSONAL FILE IS READ AND CHANGED FROM
 266  REM  ASCII CODE TO NORMAL CHARACTERS.
 285  REM (290 - 310) - THE SUBJECT'S PARAMETER FILE IS READ FROM DISK TO DMEM AND
 286  REM CHANGES FROM ASCII CODE TO NORMAL CHARACTERS.
 315  REM (320) - THE SWEEP LENGTH AND THE NUMBER OF CHANNELS RECORDED ARE
 316  REM   TAKEN FROM THE PARAMETER FILE AND THE PERSONAL FILE RESPECTFULLY.
 317  REM   THE VARIABLE WITH THE CHANNEL INFORMATION IS ALSO PLACED INTO A
 318  REM   NON-STRING VARIABLE.  THE TIME SCALE INCRIMENTATION IS DETERMINED.
 325  REM (330 - 370) - INPUT THE MIN. VOLTAGE LEVEL OF THE WAVEFORM.  USER IS
 326  REM    ASKED TO VALIDATE VARIABLE'S VALUE.  THE VOLTAGE INCRIMENTATION IS
 327  REM    DETERMINED
 375  REM (380 - 410) - INPUT THE GAIN TO BE USED IN PLOTTING.  USER IS ASKED TO
 376  REM    VALIDATE THE VARIABLE'S VALUE.
 415  REM (420) - SETTING THE POINT OF ORIGIN FOR THE ZETA PLOTTER TO THE
 416  REM   PRESENT PEN LOCATION AND A PEN ROTATION OF 1.0
 425  REM (430) - SETTING A STRING VARIABLE EQUAL TO THE SUBJECT NUMBER.
 435  REM (440) - FIND FILENAME OF THE WAVEFORMS TO BE PLOTTED.
 445  REM (450 - 480) - CONDITIONAL BRANCH FOR PLOTTING OUT THE CORRECT
 446  REM   NUMBER OF CHANNELS.
 485  REM (485 - 810) - PLOTS OUT THE ENCLOSURE FOR THE WAVEFORM AND HEADER INFO.
 815  REM (820 - 1170) - PLOTS OUT THE INDIVIDUAL EVOKED POTENTIAL WAVEFORM IN
 816  REM   IT'S ENCLOSURE.
1175  REM (1180 - 1190) - ENABLES USER TO PLOT ANOTHER FILE.
1195  REM (1200 - 1220) - ASKS IF FILE IS IN THIS SERIES.
1225  REM (1230 - 1240) - CHECKS TO SEE IF FILE IS NEXT IN SEQUENCE.
1245  REM (1250) - INCREASES THE SUBSCRIPT BY 1.  ADVANCES GRAPH PAPER, AND
1246  REM    RERUNS THE PROGRAM
1255  REM (1260 - 1300) - IF FILE IS NOT IN SEQUENCE OR IN THIS SERIES INPUT
1256  REM    THE CORRECT FILE SUBSCRIPT.  ADVANCE GRAPH PAPER AND RERUN PROG.
1305  REM (1310 - 1330) - ENABLES USER TO EXIT PROGRAM OR RUN ANOTHER PROGRAM
1335  REM END OF THE MAIN PROGRAM AND START OF SUBROUTINES.
1345  REM END OF THE MAIN PROGRAM AND START OF SUBROUTINES
1346  REM (1350 - 1380) - PLOTTING OF THE CHANNEL NUMBER AND ITS MONTAGE INFO.
1385  REM (1390 - 1420) - PLOTS OUT THE HEADER '(IN UV)' AT A RIGHT ANGLE TO
1386  REM    THE OTHER HEADER INFORMATION.
```

```
1425  REM (1430 - 1460) - IF A ODD NUMBER OF CHANNELS ARE TO BE PLOTTED THIS
1426  REM    PLOTS OUT THE TOP HORIZONTAL LINE AND INCRIMENTS THE TICK MARKS FOR
1427  REM    THE LAST CHANNEL.
1465  REM (1470 - 1730) - PLOTS OUT THE HEADER INFORMATION.
1735  REM (1740 - 1780) - PLOTS OUT THE LOWER HORIZONTAL LINE WITH ITS
1736  REM    INCRIMENTED TICK MARKS.
1785  REM (1790) - PLOTS OUT THE HEADER INFORMATION '(IN MSEC)'.
1795  REM (1800 - 1840) - PLOTS OUT THE INCRIMENTED TIME SCALE AND ITS TICK
1796  REM    MARKS.
1845  REM (1850 - 1870) - PLOTS OUT THE CENTER LINE FOR THE LOWER ENCLOSURE.
1875  REM (1880 - 1920) - PLOTS OUT THE RIGHT VERTICAL LINE AND ITS
1876  REM    INCRIMENTED TICK MARKS FOR THE LOWER RIGHT VERTICAL LINE.
1925  REM (1930 - 1970) - PLOTS OUT THE LEFT VERTICAL LINE WITH ITS
1926  REM    INCRIMENTED VOLTAGE SCALE AND THE TICK MARKS.
1975  REM (1980 - 2010) - PLOTS OUT THE TOP HORIZONTAL LINE FOR THE UPPER
1976  REM    ENCLOSURE.
2015  REM (2020 - 2040) - PLOTS OUT THE CENTER LINE FOR THE UPPER ENCLOSURE.
2045  REM (2050 - 2090) - PLOTS OUT THE UPPER RIGHT VERTICAL LINE AND ITS TICK
2046  REM    MARKS
2095  REM (2100 - 2180) - ADVANCES THE PLOTTING PAPER OF THE ZETA PLOTTER FOR THE
2096  REM    PLOTTING OF ANOTHER EVOKED POTENTIAL FILE.
2185  REM END OF THE PROGRAM
9999  END
$

OK
```

```
LIST LSTPER
$
    10  DIM Y$(225) , V$(66) , SUBNO$(4)
    20  PUT 0 IN PST ; PUT 1024 IN PSZ ; PUT 2 IN PDV
    30  PRINT; PRINT; PRINT; PRINT; PRINT; PRINT; PRINT; PRINT
    40  PRINT; INPUT "WHAT IS THE SUBJECT'S I.D. NUMBER - ", SUBNO$(1)
    50  PRINT "IS ", SUBNO$(1), " CORRECT?"
    60  INPUT X$
    70  IF X$ = "Y" THEN 80 ; IF X$ = X$ THEN 30
    80  BASE(1) = VAL(SUBNO$(3,4)) ; HEAD(1) = ASC(SUBNO$(1))
    90  PRINT; INPUT "WHAT IS THE FILE SUBSCRIPT (3 #'S 0 - 999) - ", SUB(1)
   100  PRINT "IS ", SUB(1), " CORRECT?"
   110  INPUT X$
   120  IF X$ = "Y" THEN 130 ; IF X$ = X$ THEN 90
   130  IF SUB(1) < 0 OR SUB(1) > 999 THEN 90
   140  IF HEAD(1) = 40 THEN 150 ; IF HEAD(1) = 50 THEN 160
   150  PUT 200 IN PFH ; PUT BASE(1) IN PFB ; PUT SUB(1) IN PFS ; GOTO 170
   160  PUT 210 IN PFH ; PUT BASE(1) IN PFB ; PUT SUB(1) IN PFS
   170  XEQ DRD
   180  FOR S = 1 TO 225
   190  X = DMEM(S) ; Y$(S,S) = CHR$(X)
   200  NEXT S
   210  PUT 208 IN PFH
   220  XEQ DRD
   230  FOR S = 1 TO 66
   240  X = DMEM(S) ; V$(S,S) = CHR$(X)
   250  NEXT S
   260  PRINT ; PRINT "TURN PAPER TO THE TOP OF A FRESH PAGE SO THAT ALL"
   270  PRINT "      DATA WILL BE PRINTED ON IT'S OWN PAGE."
   280  INPUT " AFTER FIXING PAPER TYPE Y TO CONTINUE.", X$
   290  IF X$ = "Y" THEN 300 ; IF X$ = X$ THEN 260
   300  PRINT; PRINT; PRINT; PRINT "SUBJECT'S NAME - ", Y$(1,34)
   310  PRINT ; PRINT "SUBJECT'S I.D. NUMBER - ", Y$(35,39)
   320  PRINT ; PRINT "DATE OF EXPERIMENT - ", Y$(41,48)
   330  PRINT ; PRINT "G-FORCE OF EXPERIMENT - ", Y$(50,55)
   340  PRINT ; PRINT "EXPERIMENT'S RUN NUMBER - ", Y$(56,61)
   350  PRINT ; PRINT "TIME-OF-DAY - ", Y$(63,68)
   360  PRINT ; PRINT "AIR TEMP IN ROOM (IN F) - ", Y$(69,73)
   370  PRINT ; PRINT "SUBJECT'S TEMP (IN F) - ", Y$(74,78)
   380  PRINT ; PRINT "SUBJECT'S HEART RATE (PER MIN) - ", Y$(79,82)
   390  PRINT ; PRINT "COMMENTS - ", Y$(83,142)
   400  PRINT ; PRINT "WRIST STIMULATED - ", V$(1,5)
   410  PRINT ; PRINT "SUBJECT'S THRESHOLD - ", V$(36,38)
   420  PRINT ; PRINT "STIMULUS INTENSITY - ", V$(40,43)
   430  PRINT ; PRINT "DELAY TIME (IN MS) - ", V$(8,9)
   440  PRINT ; PRINT "STIMULI PER SECOND - ", V$(11,12)
   450  PRINT ; PRINT "LENGTH OF SWEEP - ", V$(30,34)
   460  PRINT ; PRINT "DWELL TIME - ", V$(25,28)
   470  PRINT ; PRINT "STIMULUS PERIOD - ", V$(13,14)
   480  PRINT ; PRINT "NUMBER OF SWEEPS - ", V$(60,63)
```

```
    490  PRINT ; PRINT "NUMBER OF CHANNELS - ", Y$(143,144)
    500  PRINT; PRINT "                   MONTAGE FOR CHANNEL # 1"
    510  PRINT; PRINT "   INPUT A", Y$(215,219), "   INPUT B", Y$(220,224)
    520  PRINT; PRINT "                   MONTAGE FOR CHANNEL # 2"
    530  PRINT; PRINT "   INPUT A", Y$(205,209), "   INPUT B", Y$(210,214)
    540  PRINT ; PRINT
    550  PRINT; PRINT "                   MONTAGE FOR CHANNEL # 3"
    560  PRINT; PRINT "   INPUT A", Y$(195,199), "   INPUT B", Y$(200 - 204)
    570  PRINT; PRINT "                   MONTAGE FOR CHANNEL # 4"
    580  PRINT; PRINT "   INPUT A", Y$(185,189), "   INPUT B", Y$(190,194)
    590  PRINT; PRINT "                   MONTAGE FOR CHANNEL # 5"
    600  PRINT; PRINT "   INPUT A", Y$(175,179), "   INPUT B", Y$(180,184)
    610  PRINT; PRINT "                   MONTAGE FOR CHANNEL # 6"
    620  PRINT; PRINT "   INPUT A", Y$(165,169), "   INPUT B", Y$(170,174)
    630  PRINT; PRINT "                   MONTAGE FOR CHANNEL # 7"
    640  PRINT; PRINT "   INPUT A", Y$(155,159), "   INPUT B", Y$(160,164)
    650  PRINT; PRINT "                   MONTAGE FOR CHANNEL # 8"
    660  PRINT; PRINT "   INPUT A", Y$(145,149), "   INPUT B", Y$(150,154)
    670  PRINT; PRINT "START OF DATA MEMORY - ", "0"
    680  PRINT ; PRINT "SIZE OF DATA MEMORY - ", V$(20,24)
    690  PRINT ; PRINT "AMPLIFIER/FILTER SENSITIVITY (IN UV) - ", V$(46,49)
    700  PRINT ; PRINT "AMPLIFIER/FILTER LOW BANDPASS (IN HZ) - ", V$(51,53)
    710  PRINT ; PRINT "AMPLIFIER/FILTER HIGH BANDPASS (IN HZ) - ", V$(55,58)
    720  PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT
    730  PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT
    740  PRINT ; INPUT "DO YOU WISH TO LIST ANOTHER FILE? - ", X$
    750  IF X$ = "Y" THEN 30 ; IF X$ = "N" THEN 760 ; IF X$ = X$ THEN 740
    760  PRINT ; INPUT "DO YOU WISH TO RUN ANOTHER PROGRAM? - ", X$
    770  IF X$ = "Y" THEN 780 ; IF X$ = "N" THEN 790 ; IF X$ = X$ THEN 760
    780  RUN EPPROG
    790  STOP ; END
$

OK
```

```
LIST LISTBU
$
     5  REM DIMENSION VARIABLES
    10  DIM Y$(225) , V$(66) , SUBNO$(4)
    15  REM SET DATA MEMORY PARAMETERS AND SELECT DISK DRIVE # 2.
    20  PUT 0 IN PST ; PUT 1024 IN PSZ ; PUT 2 IN PDV
    30  PRINT; PRINT; PRINT; PRINT; PRINT; PRINT; PRINT; PRINT
    35  REM INPUT THE SUBJECT'S I.D. NUMBER
    40  PRINT; INPUT "WHAT IS THE SUBJECT'S I.D. NUMBER - ", SUBNO$(1)
    45  REM VALIDATION CHECK
    50  PRINT "IS ", SUBNO$(1), " CORRECT?"
    60  INPUT X$
    70  IF X$ = "Y" THEN 80 ; IF X$ = X$ THEN 30
    75  REM FIND THE FILE BASE AND THE FILE HEADER
    80  BASE(1) = VAL(SUBNO$(3,4)) ; HEAD(1) = ASC(SUBNO$(1))
    85  REM INPUT THE FILE SUBSCRIPT
    90  PRINT; INPUT "WHAT IS THE FILE SUBSCRIPT (3 #'S 0 - 999) - ", SUB(1)
    95  REM VALIDATION CHECK
   100  PRINT "IS ", SUB(1), " CORRECT?"
   110  INPUT X$
   120  IF X$ = "Y" THEN 130 ; IF X$ = X$ THEN 90
   125  REM CHECK THE DATA RANGE OF THE SUBSCRIPT
   130  IF SUB(1) < 0 OR SUB(1) > 999 THEN 90
   135  REM DETERMINE IF SUBJECT IS HUMAN OR RHESUS AND READ THAT PERSONAL FILE.
   140  IF HEAD(1) = 40 THEN 150 ; IF HEAD(1) = 50 THEN 160
   150  PUT 200 IN PFH ; PUT BASE(1) IN PFB ; PUT SUB(1) IN PFS ; GOTO 170
   160  PUT 210 IN PFH ; PUT BASE(1) IN PFB ; PUT SUB(1) IN PFS
   170  XEQ DRD
   175  REM TRANSULATE DATA FROM ASCII CODE TO NORMAL CHARA. AND PLACE IN VARIABLE.
   180  FOR S = 1 TO 225
   190  X = DMEM(S) ; Y$(S,S) = CHR$(X)
   200  NEXT S
   205  REM READ THE PAPAMETER FILE
   210  PUT 208 IN PFH
   220  XEQ DRD
   225  REM TRANSULATE DATA FROM ASCII CODE TO NORMAL CHARA. AND PLACE IN VARIABLE.
   230  FOR S = 1 TO 66
   240  X = DMEM(S) ; V$(S,S) = CHR$(X)
   250  NEXT S
   255  REM ADVANCE PAPER TO TOP OF A CLEAN PAGE.
   260  PRINT ; PRINT "TURN PAPER TO THE TOP OF A FRESH PAGE SO THAT ALL"
   270  PRINT "     DATA WILL BE PRINTED ON IT'S OWN PAGE."
   280  INPUT " AFTER FIXING PAPER TYPE Y TO CONTINUE.", X$
   290  IF X$ = "Y" THEN 300 ; IF X$ = X$ THEN 260
   295  REM PRINT OUT THE VARIABLES AND THEIR VALUES
   300  PRINT; PRINT; PRINT; PRINT "SUBJECT'S NAME - ", Y$(1,34)
   310  PRINT ; PRINT "SUBJECT'S I.D. NUMBER - ", Y$(35,39)
   320  PRINT ; PRINT "DATE OF EXPERIMENT - ", Y$(41,48)
   330  PRINT ; PRINT "G-FORCE OF EXPERIMENT - ", Y$(50,55)
   340  PRINT ; PRINT "EXPERIMENT'S RUN NUMBER - ", Y$(56,61)
```

```
350  PRINT ; PRINT "TIME-OF-DAY - ", Y$(63,68)
360  PRINT ; PRINT "AIR TEMP IN ROOM (IN F) - ", Y$(69,73)
370  PRINT ; PRINT "SUBJECT'S TEMP (IN F) - ", Y$(74,78)
380  PRINT ; PRINT "SUBJECT'S HEART RATE (PER MIN) - ", Y$(79,82)
390  PRINT ; PRINT "COMMENTS - ", Y$(83,142)
400  PRINT ; PRINT "WRIST STIMULATED - ", V$(1,5)
410  PRINT ; PRINT "SUBJECT'S THRESHOLD - ", V$(36,38)
420  PRINT ; PRINT "STIMULUS INTENSITY - ", V$(40,43)
430  PRINT ; PRINT "DELAY TIME (IN MS) - ", V$(8,9)
440  PRINT ; PRINT "STIMULI PER SECOND - ", V$(11,12)
450  PRINT ; PRINT "LENGTH OF SWEEP - ", V$(30,34)
460  PRINT ; PRINT "DWELL TIME - ", V$(25,28)
470  PRINT ; PRINT "STIMULUS PERIOD - ", V$(13,14)
480  PRINT ; PRINT "NUMBER OF SWEEPS - ", V$(60,63)
490  PRINT ; PRINT "NUMBER OF CHANNELS - ", Y$(143,144)
500  PRINT; PRINT "                MONTAGE FOR CHANNEL # 1"
510  PRINT; PRINT "   INPUT A", Y$(215,219), "   INPUT B", Y$(220,224)
520  PRINT; PRINT "                MONTAGE FOR CHANNEL # 2"
530  PRINT; PRINT "   INPUT A", Y$(205,209), "   INPUT B", Y$(210,214)
540  PRINT ; PRINT
550  PRINT; PRINT "                MONTAGE FOR CHANNEL # 3"
560  PRINT; PRINT "   INPUT A", Y$(195,199), "   INPUT B", Y$(200 - 204)
570  PRINT; PRINT "                MONTAGE FOR CHANNEL # 4"
580  PRINT; PRINT "   INPUT A", Y$(185,189), "   INPUT B", Y$(190,194)
590  PRINT; PRINT "                MONTAGE FOR CHANNEL # 5"
600  PRINT; PRINT "   INPUT A", Y$(175,179), "   INPUT B", Y$(180,184)
610  PRINT; PRINT "                MONTAGE FOR CHANNEL # 6"
620  PRINT; PRINT "   INPUT A", Y$(165,169), "   INPUT B", Y$(170,174)
630  PRINT; PRINT "                MONTAGE FOR CHANNEL # 7"
640  PRINT; PRINT "   INPUT A", Y$(155,159), "   INPUT B", Y$(160,164)
650  PRINT; PRINT "                MONTAGE FOR CHANNEL # 8"
660  PRINT; PRINT "   INPUT A", Y$(145,149), "   INPUT B", Y$(150,154)
670  PRINT; PRINT "START OF DATA MEMORY - ", "0"
680  PRINT ; PRINT "SIZE OF DATA MEMORY - ", V$(20,24)
690  PRINT ; PRINT "AMPLIFIER/FILTER SENSITIVITY (IN UV) - ", V$(46,49)
700  PRINT ; PRINT "AMPLIFIER/FILTER LOW BANDPASS (IN HZ) - ", V$(51,53)
710  PRINT ; PRINT "AMPLIFIER/FILTER HIGH BANDPASS (IN HZ) - ", V$(55,58)
715  REM ADVANCE PAPER
720  PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT
730  PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT ; PRINT
735  REM RE-RUN THE PROGRAM IF WANTED.
740  PRINT ; INPUT "DO YOU WISH TO LIST ANOTHER FILE? - ", X$
750  IF X$ = "Y" THEN 30 ; IF X$ = "N" THEN 760 ; IF X$ = X$ THEN 740
755  REM RUN POINTER PROGRAM IF USER WANTS TO RUN ANOTHER PROGRAM.
760  PRINT ; INPUT "DO YOU WISH TO RUN ANOTHER PROGRAM? - ", X$
770  IF X$ = "Y" THEN 780 ; IF X$ = "N" THEN 790 ; IF X$ = X$ THEN 760
780  RUN EPPROG
785  REM STOP THE PROGRAM
790  STOP ; END
$
```